

**DIGITAL IMAGE ENCRYPTION TECHNIQUES FOR WIRELESS
SENSOR NETWORKS**

BY

AHMAD MOHAMMAD SHAHEEN

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER NETWORKS

MARCH 2017

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Ahmad M. Shaheen** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS**.




Dr. Ahmad Al-Mulhem
Department Chairman



Dr. Salam A. Zummo
Dean of Graduate Studies

23/5/17

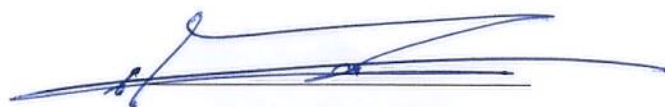
Date



Dr. Talal M. Al-Kharoubi
(Advisor)



Dr. Tarek R. Sheltami
(Member)



Dr. Basem Al-Madani
(Member)

© Ahmad Shaheen

2017

[Dedicated to My Parents, Family and Friends]

ACKNOWLEDGMENTS

All praise and glory to Allah the most merciful, the most beneficent, who gave me the health, strength, and courage to complete my Master's degree.

I would like to express my deep appreciation to my advisor Dr. Talal Al-Kharoubi for giving me the opportunity to become one of his students. I thank him for his efficient and constant support, help, motivation, and immense knowledge. His precious advice and thorough guidance played a critical role in the completion of this thesis.

I also would like to extend my appreciation to my dissertation committee members Dr. Tarek Sheltami and Dr. Basem Al-Madani for their insightful comments, support, and profitable questions which incited me to enhance my work.

I am thankful to the King Fahd University of Petroleum and Minerals (KFUPM) for providing me with the research facilities, precious resources, financial support, and an environment conducive to intellectual growth for my master thesis.

I would like to thank those with whom I have a much deeper relationship, among them are my parents, who brought me up and supported me throughout my education.

Special thanks for my elder brother Waheed Shaheen, for taking care of me and being the one who stood and still standing and encouraging me during my personal and educational life.

I also would like to thank my brothers Hamada and Sa'ed Shaheen, and my sisters for sincerely caring about my well-being and for their love and affection.

Finally, I would like to thank many other people from KFUPM who have made my years unforgettable and cherished. They include but not limited to: Bashar Khatib, Khaled Barad'ieh, Ibrahim Hussein, Mohammed Qannan and Shadi Al-Haj. I would like to thank these individuals for their friendship and motivation[

TABLE OF CONTENTS

ACKNOWLEDGMENTS	V
TABLE OF CONTENTS	VI
LIST OF TABLES.....	IX
LIST OF FIGURES.....	XI
LIST OF ABBREVIATIONS	XIV
ABSTRACT	XV
ملخص الرسالة.....	XVII
CHAPTER 1 INTRODUCTION.....	1
1.1 Discrete Cosine Transform	1
1.2 Discrete Wavelet Transform.....	6
1.3 Thesis Objectives and Methodology	8
CHAPTER 2 LITERATURE REVIEW.....	10
2.1 Cryptography	10
2.2 Encryption and Decryption.....	11
2.3 Digital Image Transformation Techniques.....	13
2.3.1 Karhunen-Loève Transform	13
2.3.2 Discrete Cosine Transform.....	13
2.3.3 Discrete Wavelet Transform	14
2.4 Digital Image Encryption Techniques	15
2.4.1 Blowfish Encryption.....	15
2.4.2 Advances Encryption Standard (Rijndael)	15

2.4.3	Chaos-Based Encryption	17
2.4.4	Hill Cipher.....	19
2.4.5	Permutation Based Encryption	19
2.4.6	Differential Evolution	20
2.4.7	RC6 Encryption	20
2.5	Summery.....	21
 CHAPTER 3 THE PROPOSED TECHNIQUES AND EXPERIMENTAL IMPLEMENTATION.....		26
3.1	Wireless Sensor Networks	26
3.2	The Main Components of WSN	28
3.3	Applications of Wireless Sensor Networks	29
3.4	Challenges in Designing a WSN	32
3.5	The Proposed Technique Using DCT	33
3.6	The Proposed Technique Using DWT.....	43
3.7	Experimental Implementation	54
3.8	Experimental Test Cases of Digital Images	54
3.9	Experimental Networks Topologies.....	57
3.9.1	Single-Hop Network	57
3.9.2	Multi-Hop Network	59
3.10	Performance Metrics for Experimental Testing	60
3.10.1	Peak Signal to Noise Ratio (PSNR)	61
3.10.2	Structural Similarity (SSIM)	62
3.10.3	Histogram Analysis	63
3.10.4	End-to-End Delay	63
3.10.5	Energy Efficiency	64

CHAPTER 4 EXPERIMENTAL RESULTS EVALUATION AND DISCUSSION.....	65
4.1 PSNR and SSIM.....	65
4.1.1 Cameraman Image using DCT Algorithm.....	65
4.1.2 Einstein Image using DCT Algorithm	68
4.1.3 Peppers Image using DCT Algorithm.....	70
4.1.4 Cameraman Image using DWT Algorithm	72
4.1.5 Einstein Image using DWT Algorithm.....	74
4.1.6 Peppers Image using DWT Algorithm	76
4.1.7 Comparison Between DCT and DWT Results.....	79
4.2 Histogram Analysis.....	81
4.3 End-to-End Delay	88
4.4 Energy Efficiency and Power Consumption	93
4.5 Comparison Between the Proposed Algorithms and Others Work	94
CHAPTER 5 CONCLUSION AND FUTURE WORK.....	95
5.1 Research Conclusion	95
5.2 Future Work.....	96
References	97
VITAE.....	102

LIST OF TABLES

Table 2.1: Image encryption techniques in literature survey.....	22
Table 3.1: Tools and Software Used for Implementation.....	54
Table 3.2: Test Cases Parameters.....	55
Table 3.3: Single-Hop Scenario Parameters.....	58
Table 3.4: Multi-Hop Scenarios Parameters.....	59
Table 4.1: PSNR and SSIM results of Cameraman Image with DCT 8x8 Block Size.....	66
Table 4.2: PSNR and SSIM results of Cameraman Image with DCT 16x16 Block Size..	66
Table 4.3: PSNR and SSIM results of Cameraman Image with DCT 32x32 Block Siz...	67
Table 4.4: PSNR and SSIM results of Einstein Image with DCT 8x8 Block Size.....	68
Table 4.5: PSNR and SSIM results of Einstein Image with DCT 16x16 Block Size.....	69
Table 4.6: PSNR and SSIM results of Einstein Image with DCT 32x32 Block Size.....	70
Table 4.7: PSNR and SSIM results of Peppers Image with DCT 8x8 Block Size.....	70
Table 4.8: PSNR and SSIM results of Peppers Image with DCT 16x16 Block Size.....	71
Table 4.9: PSNR and SSIM results of Peppers Image with DCT 32x32 Block Size.....	72
Table 4.10: PSNR and SSIM results of Cameraman Image with DWT 8x8 Block Size..	72
Table 4.11: PSNR and SSIM results of Cameraman Image with DWT 16x16 Block Size.....	73
Table 4.12: PSNR and SSIM results of Cameraman Image with DWT 32x32 Block Size.....	74
Table 4.13: PSNR and SSIM results of Einstein Image with DWT 8x8 Block Size.....	74
Table 4.14: PSNR and SSIM results of Einstein Image with DWT 16x16 Block Size....	75
Table 4.15: PSNR and SSIM results of Einstein Image with DWT 32x32 Block Size....	76
Table 4.16: PSNR and SSIM results of Peppers Image with DWT 8x8 Block Size.....	76
Table 4.17: PSNR and SSIM results of Peppers Image with DWT 16x16 Block Size....	77

Table 4.18: PSNR and SSIM results of Peppers Image with DWT 32x32 Block Size....	78
Table 4.19: PSNR Values of the Proposed Algorithms of the Original and Encrypted Images.....	79
Table 4.20: SSIM Values of the Proposed Algorithms of the Original and Encrypted Images.....	80
Table 4.21: The Average Time for DCT and DWT.....	88
Table 4.22: The Total End-to-End Delay for DCT Through Single-Hop Network.....	89
Table 4.23: The Total End-to-End Delay for DWT Through Single-Hop Network.....	90
Table 4.24: The Total End-to-End Delay for DCT Through Multi-Hop Network.....	91
Table 4.25: The Total End-to-End Delay for DWT Through Multi-Hop Network.....	92
Table 4.26: Comparison Between the Proposed Algorithms and Other Algorithms.....	94

LIST OF FIGURES

Figure 1.1: 8×8 Block DCTs Basis Patterns.....	4
Figure 1.2: Low to High Zigzag order of the sequence of an 8×8 frequency block.....	5
Figure 1.3: The “cameraman” image with different cut-off frequencies.....	6
Figure 1.4 (a): Two dimensional 2-levels DWT decomposition.....	7
Figure 1.4 (b): The original standard "Tree.tiff" image.....	7
Figure 1.4 (c): Two levels DWT decomposition of the standard Tree.tiff image.....	7
Figure 2.1: The process of image encryption and decryption.....	13
Figure 3.1: The Structure of Wireless Sensor Networks.....	27
Figure 3.2: Wireless Sensor Networks Applications.....	31
Figure 3.3: General Overview of The Encryption and Decryption using DCT.....	33
Figure 3.4: The Distribution of Key Matrices.....	34
Figure 3.5: DCT Encryption Algorithm Flowchart.....	37
Figure 3.6: DCT Decryption Algorithm Flowchart.....	41
Figure 3.7: General Overview of The Encryption and Decryption using DWT.....	43
Figure 3.8: DWT Encryption Algorithm Flowchart.....	47
Figure 3.9: DWT Decryption Algorithm Flowchart.....	52
Figure 3.10: Cameraman Image.....	56
Figure 3.11: Einstein Image.....	56
Figure 3.12: Peppers Image.....	57
Figure 3.13: Single-Hop Simulation.....	58
Figure 3.14: Multi-Hop Simulation.....	60
Figure 4.1: Encrypted and Decrypted Cameraman Image using DCT and 8x8 Block Size.....	65

Figure 4.2: Encrypted and Decrypted Cameraman Image using DCT and 16x16 Block Size.....	66
Figure 4.3: Encrypted and Decrypted Cameraman Image using DCT and 32x32 Block Size.....	67
Figure 4.4: Encrypted and Decrypted Einstein Image using DCT and 8x8 Block Size....	68
Figure 4.5: Encrypted and Decrypted Einstein Image using DCT and 16x16 Block Siz..	69
Figure 4.6: Encrypted and Decrypted Einstein Image using DCT and 32x32 Block Size.....	69
Figure 4.7: Encrypted and Decrypted Peppers Image using DCT and 8x8 Block Size....	70
Figure 4.8: Encrypted and Decrypted Peppers Image using DCT and 16x16 Block Size..	71
Figure 4.9: Encrypted and Decrypted Peppers Image using DCT and 32x32 Block Size..	71
Figure 4.10: Encrypted and Decrypted Cameraman Image using DWT and 8x8 Block Size.....	72
Figure 4.11: Encrypted and Decrypted Cameraman Image using DWT and 16x16 Block Size.....	73
Figure 4.12: Encrypted and Decrypted Cameraman Image using DWT and 32x32 Block Size.....	73
Figure 4.13: Encrypted and Decrypted Einstein Image using DWT and 8x8 Block Size.....	74
Figure 4.14: Encrypted and Decrypted Einstein Image using DWT and 16x16 Block Size.....	75
Figure 4.15: Encrypted and Decrypted Einstein Image using DWT and 32x32 Block Size.....	75
Figure 4.16: Encrypted and Decrypted Peppers Image using DWT and 8x8 Block Size..	76
Figure 4.17: Encrypted and Decrypted Peppers Image using DWT and 16x16 Block Size.....	77
Figure 4.18: Encrypted and Decrypted Peppers Image using DWT and 32x32 Block Size.....	77
Figure 4.19: PSNR Values of the Proposed Algorithms of the Original and Encrypted Images.....	80

Figure 4.20: SSIM Values of the Proposed Algorithms of the Original and Encrypted Images.....	81
Figure 4.21: Histogram Analysis of DCT Algorithm with 8x8 Block Size.....	82
Figure 4.22: Histogram Analysis of DCT Algorithm with 16x16 Block Size.....	83
Figure 4.23: Histogram Analysis of DCT Algorithm with 32x32 Block Size.....	84
Figure 4.24: Histogram Analysis of DWT Algorithm with 8x8 Block Size.....	85
Figure 4.25: Histogram Analysis of DWT Algorithm with 16x16 Block Size.....	86
Figure 4.26: Histogram Analysis of DWT Algorithm with 32x32 Block Size.....	87

LIST OF ABBREVIATIONS]

DCT	:	Discrete Cosine Transform
DWT	:	Discrete Wavelet Transform
WSN	:	Wireless Sensor Network
AES	:	Advanced Encryption Standard
SSIM	:	Structural Similarity
PSNR	:	Peak Signal to Noise Ratio
QoS	:	Quality of Service
E-E Delay	:	End to End Delay

ABSTRACT

Full Name : [Ahmad Mohammad Abdel Rahman Shaheen]
Thesis Title : [Digital Image Encryption Techniques for Wireless Sensor Networks]
Major Field : [Computer Networks]
Date of Degree : [March 2017]

Confidentiality in storing and transmitting images is needed for different fields such as medical, military, online personal albums, confidential communications and video conferencing...etc. Many image encryption techniques are proposed to ensure confidentiality of data. Digital images are different from text data as they have more data, higher data redundancy and correlation between image pixels.

In Wireless Sensor Networks (WSN), many encryption techniques are proposed. Sensor nodes have limited resources in memory, energy and processing capabilities. Any proposed technique must consider these limitations. However, most of the proposed techniques are not applicable for digital images due to image structure and size, so the traditional cryptosystems can't be applied on WSN.

In this thesis, the digital images transformation techniques: Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) are used to propose digital images encryption techniques for WSN.

The proposed techniques were implemented using Matlab, and tested on WSN using Contiki OS and its simulator Cooja. Both techniques were tested using several performance metrics such as PSNR, SSIM, Histogram Analysis and End to End Delay.

The proposed algorithms provide good results in terms of PSNR, SSIM and histogram analysis. In terms of End to End delay and Energy consumption the proposed DWT algorithm was better than DCT because it needs less time of computation. In general, the proposed DWT was better than the proposed DCT.

ملخص الرسالة

الاسم الكامل: أحمد محمد عبد الرحمن شاهين

عنوان الرسالة: تقنيات تشفير الصور الرقمية لشبكات الاستشعار اللاسلكية

التخصص: شبكات الحاسوب

تاريخ الدرجة العلمية: جمادى الآخرة 1438

الحفاظ على سرية تخزين و إرسال الصور الرقمية مطلب أساسي في عدة مجالات, مثل استخدام الصور الطبية الخاصة بالمريض و الصور العسكرية و الصور الشخصية المحفوظة على خوادم الشبكة و غيرها العديد من مجالات استخدام الصور الرقمية. هناك العديد من تقنيات تشفير الصور الرقمية التي تم طرحها للحفاظ على أمن و سرية الصور الرقمية. تختلف الصور الرقمية في طبيعتها عن الأنواع الأخرى من البيانات مثل النصوص, حيث ان الصور الرقمية تحتوي على كمية أكبر من البيانات, نسبة أعلى من التكرارات في البيانات نفسها, وايضاً هناك علاقة ترابط بين محتوى البيانات في الصور الرقمية.

في شبكات الاستشعار اللاسلكية, تم طرح العديد من تقنيات تشفير البيانات. ولكن نقطة الاستشعار في شبكة الاستشعار اللاسلكية لها خصائص معينة مثل محدودية قدرات التخزين و المعالجة والطاقة ولهذا فان هذه التقنيات يجب ان تأخذ بعين الاعتبار محدودية قدرات نقاط الاستشعار. و على الرغم من ذلك, فان معظم تقنيات تشفير البيانات المطروحة غير مناسبة للاستخدام في تشفير الصور الرقمية و ذلك بسبب طبيعة بنية الصورة و حجمها, ولذلك فان هذه التقنيات لا تصلح للاستخدام لتشفير الصور الرقمية في شبكات الاستشعار اللاسلكية.

في هذا البحث, تم استخدام تقنيات التحويل DCT و DWT لطرح تقنيات تشفير الصور الرقمية في شبكات الاستشعار اللاسلكية. تم تصميم و تطبيق هذه التقنيات باستخدام Matlab و تمت تجربتها على شبكة الاستشعار اللاسلكية باستخدام نظام التشغيل Contiki والمحاكي الخاص به المسمى ب Cooja. و تم فحص جودة الخوارزميات المطروحة باستخدام عدة مقاييس وهي PSNR و SSIM و تحليل الرسم البياني لتوزيع الالوان و الوقت اللازم لمعالجة و نقل الصور.

خوارزمية DCT المطروحة اعطت نتائج جيدة من ناحية ال PSNR و SSIM و الرسم البياني لتوزيع الالوان. اما من ناحية معدل التأخير فان خوارزمية DWT المطروحة اعطت نتائج افضل من خوارزمية DCT المطروحة و يعود ذلك لسبب وجود عمليات حسابية اقل فيها. اما بشكل عام, فان الخوارزمية المطروحة باستخدام نظام التحويل DWT تعد افضل من خوارزمية DCT المطروحة, وكلاهما كانت نتائجهم مرضية ومتقاربة جداً.

CHAPTER 1

INTRODUCTION

Data encryption techniques are known for a very long time. Those techniques can be used in different areas to protect data from being accessed by unauthorized persons. All type of data needs to be protected when it contains some sensitive information.

Many compression techniques are proposed to transform digital images from pixel domain to frequency domain such as the Discrete Cosine Transform (DCT) [1], Discrete Wavelet Transform (DWT) [2] and the Karhunen-Loève Transform (KLT) [3]. Both DCT and DWT is intended to be used as the main techniques for this thesis.

Two different technologies can be used for data privacy in digital images. The first method is based on protecting the data through encryption techniques. In this method, the decryption of data needs a key. The second method is Steganography which is a technique for hiding a data into an image. The processing time can be reduced by compressing the image before its being encrypted. So, is important to perform image encryption and compression simultaneously [4].

The objective of this work is to propose a solution on how to protect the digital images from being exposed by unauthorized people during its transmission via WSN.

1.1 Discrete Cosine Transform

DCT was proposed first by Ahmed, N., T. Natarajan, and K. R. Rao in 1974 [1]. The DCT transformation process is defined by the equation: $Z = WBWT$. Where Z is the transformed

matrix block, W is the coefficient matrix and B is the original matrix block. DCT creates a matrix consists of NxN transformed block (Z) from the original NxN pixel samples block B. The DCT inverse process (IDCT) can be represented by the equation: $B = WTZW$.

The elements of W are:

$$W_{ij} = C_i \cos \left(\frac{(2j+1)i\pi}{2N} \right) \quad (1.1)$$

Where:

$$C_i = \sqrt{\frac{1}{N}} \quad (i = 0) \quad (1.2)$$

$$C_i = \sqrt{\frac{2}{N}} \quad (i \geq 1) \quad (1.3)$$

The coefficient matrix W is independent from the original image pixel block. So, it can be calculated based on the block size not based on the image data. Because of that it can be calculated previously and stored to be used later.

For example, a 4x4 transformation matrix values (W) =

$$\begin{bmatrix} \frac{1}{2} \cos(0) & \frac{1}{2} \cos(0) & \frac{1}{2} \cos(0) & \frac{1}{2} \cos(0) \\ \sqrt{\frac{1}{2}} \cos\left(\frac{\pi}{8}\right) & \sqrt{\frac{1}{2}} \cos\left(\frac{3\pi}{8}\right) & \sqrt{\frac{1}{2}} \cos\left(\frac{5\pi}{8}\right) & \sqrt{\frac{1}{2}} \cos\left(\frac{7\pi}{8}\right) \\ \sqrt{\frac{1}{2}} \cos\left(\frac{2\pi}{8}\right) & \sqrt{\frac{1}{2}} \cos\left(\frac{6\pi}{8}\right) & \sqrt{\frac{1}{2}} \cos\left(\frac{10\pi}{8}\right) & \sqrt{\frac{1}{2}} \cos\left(\frac{14\pi}{8}\right) \\ \sqrt{\frac{1}{2}} \cos\left(\frac{3\pi}{8}\right) & \sqrt{\frac{1}{2}} \cos\left(\frac{9\pi}{8}\right) & \sqrt{\frac{1}{2}} \cos\left(\frac{15\pi}{8}\right) & \sqrt{\frac{1}{2}} \cos\left(\frac{21\pi}{8}\right) \end{bmatrix}$$

Mathematically, the cosine function is symmetrical and repeats every 2π radians, So W can be represented by:

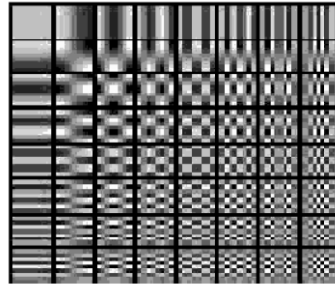
$$\begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right) & -\sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right) & -\sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right) \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right) & -\sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right) & -\sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right) \end{bmatrix}$$

After calculating the cosine values, the coefficient matrix W =

$$\begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.653 & 0.271 & 0.271 & -0.653 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.271 & -0.653 & -0.653 & 0.271 \end{bmatrix}$$

The results of a two dimensions DCT is a matrix of N by N coefficients which represents the block values of the original/plain image in the Discrete Cosine Transform domain. These coefficients could be demonstrated as weights of a set or group of standard basis patterns.

An example of 8×8 basis patterns for an 8 by 8 block image is shown in Figure 1.1. Those basis pattern are composed of a set of cosine functions. A construction of a block can be done by joining the N by N basis pattern while each basis multiplied by a different appropriate weight [5].



a) DCT transformation frequencies for 8×8 block.

DC Value (lowest Frequency)	Low Frequency	Low Frequency	High Frequency
Low Frequency	Low Frequency	Low Frequency	High Frequency
Low Frequency	Low Frequency	High Frequency	High Frequency
High Frequency	High Frequency	High Frequency	High Frequency

b) A sample of a cut off frequency by $\frac{1}{2}$ for a 4×4 block size.

DC Value (lowest Frequency)	Low Frequency	Low Frequency	Low Frequency
Low Frequency	Low Frequency	Low Frequency	High Frequency
Low Frequency	Low Frequency	Low Frequency	High Frequency
Low Frequency	High Frequency	High Frequency	High Frequency

c) A sample of a cut off frequency by $\frac{1}{4}$ for a 4×4 block size.



d) The benchmark cameraman image restored without the DC value.

Figure 1.1: 8×8 Block DCTs Basis Patterns

The frequencies order shown in Figure 1.2 will be gained after applying the DCT on a pixel domain block. The DC value is the lowest frequency gained after the DCT and it's located at the position (0, 0). In the transformed block, the DC value is considered as the most important value because it represents the general view of the block. The rest of the frequencies are called the AC values and they are sorted in a zigzag order from low to high frequencies. The block details are represented from general to more specific details as we move from low to high frequencies.

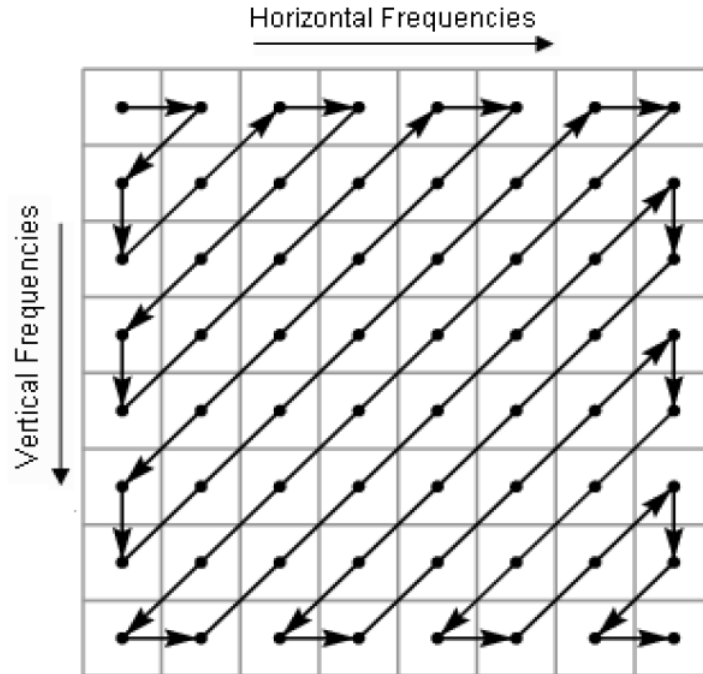


Figure 1.2 : Low to High Zigzag order of the sequence of an 8×8 frequency block [5]

For compression purposes and in order to save storage space, some of the AC values can be ignored because they are not very critical as the human vision system will not detect the distortion caused by this when transforming back to pixel domain. Figure 1.3-a show the original image. Figure 1.3-b and Figure 1.3-c show cutting off parts of the AC frequencies as on the pattern in Figure 1.1-b and Figure 1.1-c. We notice that cutting off half of AC frequencies is not noticed by the human eye, even if we removed all AC values and restored the image just by using the DC value the image still can be understood by the human eye. Figure 1.3-d show the image when its reconstructed using DC value only [5].

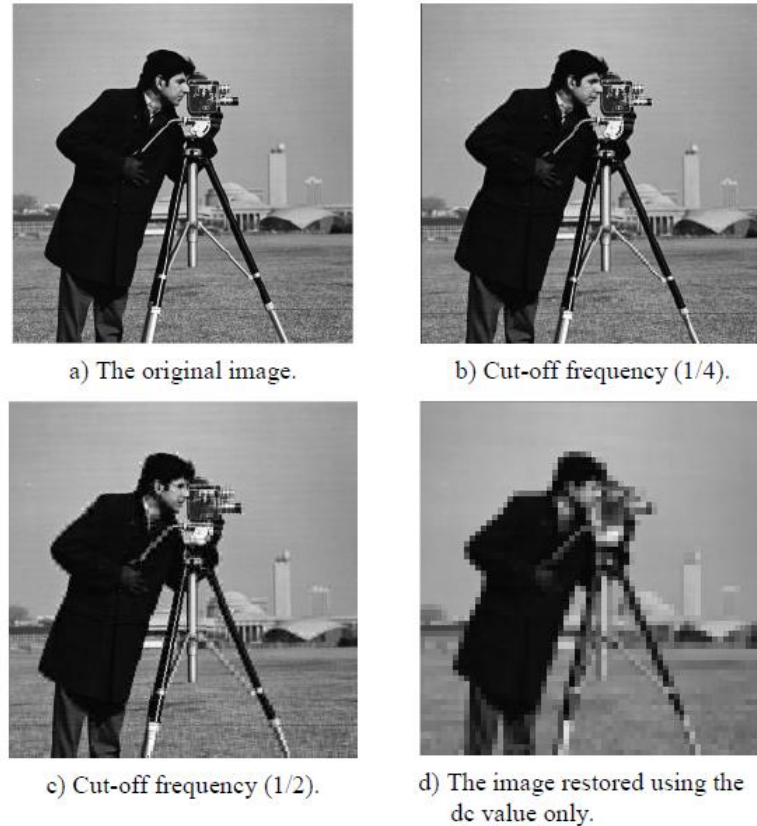


Figure 1.3 : The “cameraman” image with different cut-off frequencies [5]

1.2 Discrete Wavelet Transform

DWT (Haar Wavelet) was first introduced by Alfred Haar in 1909 [2]. The operation of this technique can be described as follows: Each pixel value of an image is filtered using: low-pass and a high-pass filters. A down sampling by factor of two is done on the results of each filter to get L and H images. After which, each column is filtered with low and high pass filters (LH, HH) and sampled down by factor of two to generate 4 sub-images (LL, LH, HL and HH) as shown in figure 1.4-a. These four sub-band images can be merged to generate an image with the same number of samples as the original one. ‘LL’ is the original image, low-pass filter applied in horizontal and vertical directions and sub-sampled by a

factor of 2. 'HL' is high-pass filter applied in the vertical direction and consists of error vertical frequencies, 'LH' is high-pass filter applied in the horizontal direction and consists of error horizontal frequencies. While 'HH' sub-band is high-pass filter applied in both directions: horizontal and vertical [2].

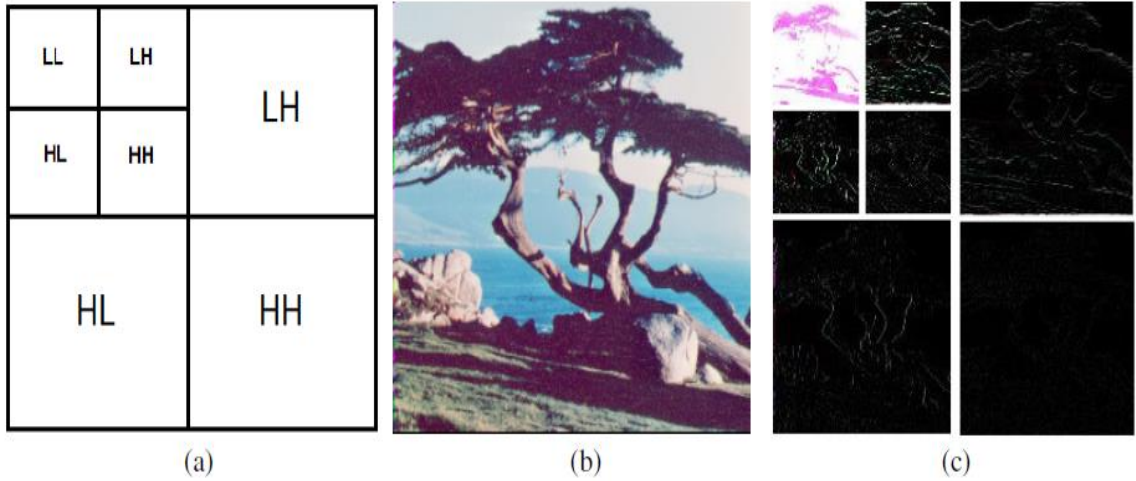


Figure 1.4: (a) Two dimensional 2-levels DWT decomposition. (b) The original standard "Tree.tiff" image. (c) Two levels DWT decomposition of the standard Tree.tiff image. Corresponding frequency locations are illustrated in (a)

The two-dimensional wavelet decomposition illustrated in Figure 1.4(a,c) is applied to the 'LL' sub-band, giving a new four sub-band images. Figure 1.4-c shows the results of two stages of wavelet decomposition when its applied on a sample image as shown in figure 1.4-b.

The Haar wavelet [2] transform can be expressed by $A = HBH^T$, where B is an $N \times N$ original image matrix, H is an $N \times N$ Haar transformation matrix, and A is the resulting $N \times N$ transform that contains the Haar basis functions, $h_k(Z)$, which are defined over the interval $Z \in [0,1]$ for $k = 0, 1, 2, \dots, N-1$, where $N = 2^n$. In order to generate H , k is defined as $k = 2^p + q - 1$, where $0 \leq p \leq n-1$, $q = 0$ or 1 for $p = 0$, and $1 \leq q \leq 2^p$ for $p \neq 0$ [2].

The Haar basis functions is:

$$h_0(Z) = h_{00}(z) = 1/\sqrt{N}, Z \in [0,1] \quad (1.4)$$

$$h_k(Z) = h_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2} & \frac{q-1}{2^p} \leq z \leq \frac{1-0.5}{2^p} \\ -2^{p/2} & \frac{q-0.5}{2^p} \leq z \leq \frac{q}{2^p} \\ 0 & \text{otherwise} \end{cases} \quad (1.5)$$

The i^{th} row of an $N \times N$ Haar transformation matrix contains the elements of $h_i(Z)$ for

$Z = 0/N, 1/N, 2/N, \dots, (N-1)/N$. [2].

An example of a 4×4 Haar transformation matrix follows:

$$H_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

1.3 Thesis Objectives and Methodology

This thesis aims to develop digital images encryption/decryption algorithms using the DCT and DWT that is suitable to transmit images over WSN. Sensitive Digital images can be encrypted by the proposed algorithms to hide all of its critical features. When it is received by the intended recipient, the encrypted image will be decrypted by reversing back the encryption process.

This thesis consists of the following phases:

- Literature review: studying and analyzing different techniques for image encryption.

- Design: Designing the encryption and decryption algorithms.
- Implementation: Implementing the encryption and decryption algorithms.
- Evaluation and Testing: Evaluating the efficiency of the proposed algorithms. To evaluate the hardness and how much easy to break the proposed encryption technique, a standard benchmark images will be used for testing. Those images are imported from the USC-SIPI (University of Southern California – Signal and Image Processing Institute) image database. Also, a comparison will be done between the proposed work and other encryption algorithms

]

Chapter 2

Literature Review

Data encryption techniques are known since a very long time. These techniques can be used in different areas to protect data. This chapter is to study the secured and reliable techniques that protect digital images from being exposed, recognized or altered. Many techniques have been proposed to transform the digital image from the pixel to frequency domain such as the Karhunen-Loève Transform (KLT), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). The main intention of this thesis is to study the highly secured digital image encryption techniques, which makes the encrypted image having no valuable information.

2.1 Cryptography

Cryptography or encryption is a method for keeping and transmitting sensitive data in a way that the authorized persons only can access and process. It is an effective way to protect the information when its stored or transmitted over any network. At the destination side, the encrypted information will be decrypted back to restore the original data. Encryption have been used in securing data such as securing networks connections, protecting text data saved on media, protecting telecommunications systems, protecting videos and images.

The use of multimedia over the internet has grown dramatically in last few years. Securing Images and videos is very important in several applications, such as video monitoring, medical and military applications. Nowadays, the transmission of multimedia is a daily

routine and it is important to find an efficient method to transmit them in a secure manner [4].

The origin of the word encryption comes from the word ‘kryptos’ which is a Greek word, which means secret or hidden. In its earliest structure, people tried to hide certain data that they wanted to be known by others by substituting parts of the data with numbers, symbols and pictures. For many reasons people, have been interested in securing their messages. The Assyrians were interested in securing their manufacturing of the pottery trade secret. The Chinese also were interested in securing their silk manufacturing trade secret. Also, the Germans were interested in securing their military secrets by using their well-known Enigma machine.

2.2 Encryption and Decryption

Encryption is a technique for converting the data into a form that cannot be understood easily by unauthorized people. Decryption is a technique to converting the encrypted data back to its original form, so it can be easily understood. Using of the encryption and decryption techniques started at beginning of using the communications. Some simple ciphers techniques were developed which includes the substitution of letters, reordering the letters in the message. More complex cipher techniques were developed to work according to advanced computer algorithms.

To restore the contents of an encrypted image, data or signal, the correct key is needed for decryption. The decryption is an algorithm that reverse the encryption algorithm using the decryption key. Figure 2.1 shows a simple diagram of encryption and decryption. Encryption/Decryption is very important in the field of image processing, because that

some sensitive images could be easily exposed. Also, encryption and decryption is a good routine when transmitting any kind of images that contains sensitive data, such as medical, military, aerospace and satellite images. The stronger encryption algorithm we use, the harder for unauthorized people to break. Strong encryption means that the ciphers are unbreakable without the using of decryption keys. While most of the companies and their customers view the encryption as a method of keeping a secured data and minimizing attacks. Some governments view the encryption as a way by which eavesdroppers might use to evade authorities. These governments, want to setup a trusted arrangement. Which means anyone who uses a cipher must provide the government with a copy of the decryption key. Decryption keys must be stored in a highly-secured place, which is only used by the authorities. Opponents of the key-screw scheme claim that criminals can hack into the scheme database and illegally steal, obtain, or alter the stored decryption keys. Supporters argue that while this is a possible way for the criminals but implementing such scheme would be better to protect encryption/decryption keys from being used freely by criminals.

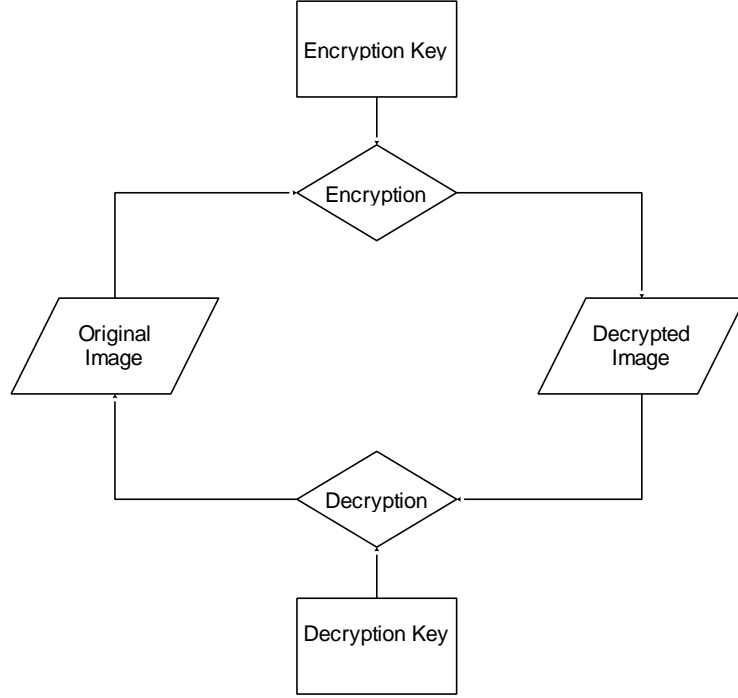


Figure 2.1: The process of image encryption and decryption

2.3 Digital Image Transformation Techniques

In this section, several image transformation techniques will be shown.

2.3.1 Karhunen-Loève Transform

In [3] the authors proposed a digital image encryption technique using the Karhunen-Loève Transform. In the proposed technique, the original image is the input of KLT encryption algorithm, to produce encrypted image and coefficient matrix. The transformed image is considered encrypted where the coefficient matrix is its decryption key. Then, RSA is used to encrypt the coefficient matrix using the receiver public key.

2.3.2 Discrete Cosine Transform

In [5] authors proposed a lossless algorithm for image encryption. The proposed technique uses the DC transform to convert the plain image from pixel to frequency domain. After the transformation process is done, the DC value in each block would be scattered. And

then, the algorithm shuffles the frequencies and then invert the sign of the block frequencies before its being transformed back using inverse function from frequency to pixel domain. The decryption algorithm returns the image back to its original form by reversing back the encryption process.

In [6] an image encryption technique using the Discrete Parametric Cosine Transform (DPCS) was proposed. The proposed method transforms the image from pixel to frequency domain using some parameters of DPCT, then using different set of parameters the (inverse of DPCT) is used to obtain the encrypted image.

In [7] the authors proposed an encryption and decryption technique using the DCT. In the encryption process, they applied the DCT on each image block then rotated the block, after that they embedded the original image with random image. At the receiver side the extracted random image is reversed the and applied the IDCT to reconstruct the original image back.

In [8] they proposed an image encryption technique using DCT and DMPFRFT. Multiple images are transformed to frequency domain using DCT, then they are processed by a filter. After that they are all multiplexed into one image and performed the DCT inverse. In the pixel domain, the pixels are scrambled and then a mask of DMPFRFT is applied to get the encrypted image. The decryption process is just the inverse of the encryption.

2.3.3 Discrete Wavelet Transform

In [2] the authors proposed a new symmetric key encryption and decryption algorithm. In this technique, the plain image is transformed from pixel to frequency domain using the Haar wavelet transform or DWT. Then, the lowest frequencies are scattered. After that, the

frequencies are shuffled and their sign are inverted, the last step is inversing back the DWT to get the encrypted image. The decryption algorithm returns the image back to its original form by reversing back the encryption process.

2.4 Digital Image Encryption Techniques

In this section, several image encryption techniques will be shown.

2.4.1 Blowfish Encryption

In [9] the authors proposed a hybrid block-based algorithm which is based on both ‘image transformation techniques’ and ‘Blowfish encryption & decryption algorithm’. The target image is divided into blocks, then each block is transformed using transformation method. After that, the image will be encrypted by the Blowfish encryption algorithm. The results showed a significant decrease in the correlation between image elements. Also, their results showed that a higher entropy and lower correlation can be gained when increasing the blocks sizes.

2.4.2 Advances Encryption Standard (Rijndael)

In [10] the authors proposed a new hybrid permutation algorithm based on both image permutation and Rijndael algorithm. The target image is splitted into 4×4 pixels blocks, and then using a permutation process the blocks are rearranged. After that, using Rijndael algorithm the image is encrypted.

In [11] authors proposed modification to the Advanced Encryption Standard algorithm (AES). A key stream generator is added to the Advanced Encryption Standard in order to ensure encryption performance improvement by reducing the entropy. The implementation of this algorithm has been done for experimental purposes. More detailed results are given

in terms of implementation and security analysis. A comparison between traditional encryption techniques and the modified AES was done to show the superiority of the modified AES technique.

In [12] the authors introduced a standard to study the spread S-boxes and analyze their strengths and weaknesses to find if they are suitable for image encryption applications. The proposed technique uses analysis results from contrast, correlation, energy, entropy, homogeneity, and mean of absolute deviation. The majority logic technique is used to decide the suitability of an S-box to image encryption applications.

In [13] Authors proposed a new digital image encryption technique based on the rotation of the Magic Cube faces. The original image is splitted into six sub images. Each sub image is divided into blocks, then attached to the faces of a Magic Cube. The magic cube is randomly rotated to mix the faces. Then result is entered to the AES encryption algorithm which is applied to the image pixels. Experimental tests and security analysis show that the proposed scheme can encrypts the image to achieve better confidentiality and can also resist exhaustive, statistical and differential attacks.

In [14] the authors proposed a modified Advanced Encryption Standard (MAES) to be used for digital image encryption, the proposed technique provides a highly-secured image encryption technique. The original AES is modified by adjusting the Shift-Row phase. Experimental studies proved the efficiency and reliability of the proposed technique compared to the original AES.

2.4.3 Chaos-Based Encryption

In [15] the authors proposed a new encryption algorithm using chaos maps and stream cipher. Two chaotic maps and a 104-bit size secret key were used for image encryption. They employed chaotic maps and the key to destroy the correlation between the cipher and the original image.

In [16] authors introduced a novel technique using both chaotic maps and genetic algorithm for digital image encryption. A set of images are generated using the original image by using the chaotic maps. Then, these generated images are used for running the genetic algorithm process. The encrypted image is also enhanced using genetic algorithm to produce the final encrypted image.

In [17] authors proposed a new encryption technique using four different chaotic maps, those four maps are compared and then some noise added on the image. First, they encrypt the original image. After that, they applied some noise on the encrypted image then decrypt it back to its original form. The Simulation results prove that Cross Chaotic map has the best results among the other four selected maps.

In [18] the authors introduced a new image encryption technique using a new chaotic system. the proposed technique work by adding two different chaotic systems together: The Rössler chaotic system and the Lorenz chaotic system. The main purpose of this algorithm is getting a stronger security.

In [19] the authors proposed a chaos based image encryption technique. The proposed algorithm is based on pixel mixing, where chaos randomness is used to scramble pixel

positions. The position of the pixels is mixed based on the randomness of the elements obtained from the chaotic map. For decryption, the pixels are arranged back to its original form. The proposed algorithm is examined using two different maps. Performance analysis is done to select the best suitable map for encryption.

In [20] the authors proposed a new image encryption technique based on 4D chaotic system. Diffusion function is only considered in the proposed technique. In order to enlarge the key space, three control parameters are added to the 4D chaotic system parameters. The control parameters are related to the original image. By three rounds of iteration, the experiments showed that the new technique can have good efficiency, fast performance and high security.

In [21] the authors proposed a novel digital image encryption technique based on chaos theory. The proposed technique relies on finding combinations between the two adjacent pixels in order to create linear independence relationships in the same row. The keys will be saved in the first column in an encrypted way using a logistic map with initial condition which is known as Key 1. Then, another key called Key2 will be used for the logistic map to change positions by shuffling them. The technique proved that it's hard enough against different attacks and it is sensitive to the initial conditions.

In [22] the authors proposed a digital image encryption technique based on chaotic sequences. The Encryption and decryption keys are acquired by 1-D Logistic map which generates the secret key for the input of the nonlinear function. The receiver can decrypt the image using the received encrypted image and the identical key sequences by inverting

the encryption process. The results of system simulations proved that the transmitted encrypted image can be reliably and correctly decrypted using the proposed technique.

In [23] the authors proposed a digital image encryption technique. Based on hierarchical combination of three chaotic maps the author proposed n-ary key stream generator. He proved that the key streams have good statistical properties. The image encryption system is implemented using the proposed approach. This technique proved that it's secure enough against different attacks.

2.4.4 Hill Cipher

In [24] the authors proposed a new image encryption algorithm using the Hill cipher technique. The proposed method will produce a self-invertible matrix. The method reduces the computational complexity of finding matrix inverse during the decryption.

In [25] the authors proposed a new modified advanced Hill encryption algorithm (AdvHill), which uses a mandatory key matrix. The proposed scheme is considered as a fast encryption scheme which avoids the problems of homogeneous background images during the encryption process. The results proved that the proposed algorithm is more durable and reliable than the original Hill encryption.

2.4.5 Permutation Based Encryption

In [26] the authors introduced a new digital image encryption technique which is based on a randomly generated pixel permutation, with emphasize to preserve the image quality. In the encryption process a key value of 64bit are maintained and sent to the receiver. The

original image is used to generate secret shares. For decryption, the receiver uses both the key and the shares to get the image.

2.4.6 Differential Evolution

In [27] the authors proposed a new technique for image encryption using the Differential Evolution approach. The work depends on deploying the discrete Fourier transform (DFT), then for encryption process the differential evolution operations will be deployed. A secret key is shared between both sender and receiver. First, 2D keyed DFT is applied on the original image. Two components are taken from the frequency domain. Then, crossover is done between two components selected based on Linear Feedback Shift Register index generator (LFSR). Also, A key mutation is applied on the real parts of a components selected based on (LFSR) index generator. The LFSR index generator initializes its seed with the shared secret key to ensure the confidentiality of the resulting indices. The encryption process shuffles the image pixels. The decryption is done using the same key in inverse process of encryption. The encrypted image is fully distorted, resulting in increasing the hardness of breaking the technique.

2.4.7 RC6 Encryption

In [28] the authors studied the efficiency of using RC6 encryption on digital images. The hardness of breaking the RC6 block encryption is also studied using different parameters like: key length, number of rounds, word size and the best choices design parameters. The analysis of RC6 block cipher security is studied for digital images. Experiments studies proved that RC6 encryption is highly reliable and secure for real time digital image encryption.

2.5 Summery

Table 2.1 shows a summary of the proposed image encryption/decryption techniques in the literature.

Table 2.1: Image encryption techniques in literature survey:

REF	Main Technique	Proposed Work/Algorithm	Main Finding
[3]	Karhunen-Loève Transform	They applied the KLT to get the transformed image and the coefficient matrix (decryption key matrix), then they encrypted the coefficient matrix using RSA algorithm.	<ul style="list-style-type: none"> - Used in frequency domain. - Uses KLT to transform. - Uses RSA to encrypt.
[5]	Discrete Cosine Transform	A technique that uses the DC transform to get the original image the frequency domain, then they applied some functions to hide the main features of the image to encrypt it.	<ul style="list-style-type: none"> - Used in frequency domain. - Uses DCT to transform. - Lossless image encryption.
[6]	Discrete Parametric Cosine Transform	A technique using DPCT and IDPCT to encrypt the image using two different sets of parameters.	<ul style="list-style-type: none"> - Uses DPCT to transform. - Uses IDPCT using different set to get the encrypted image.
[7]	Discrete Cosine Transform	A technique which transforms the image into frequencies by DCT then a random image is embedded.	<ul style="list-style-type: none"> - Used in frequency domain. - Uses DCT to transform. - Random image embedded.
[8]	Discrete Cosine Transform	They used many images to be transformed using DCT and by multiplexing the transformed images together they got one image, then IDCT is performed. In pixel the domain the pixels are scrambled and then masked by DMPFRFT to get the encrypted image.	<ul style="list-style-type: none"> - Used in frequency domain. - Uses DCT to transform. - Scrambling in the frequency domain. - Uses DMPFRFT for masking.
[2]	Discrete Wavelet Transform	A technique that uses the DWT to transform the original image into frequencies, then they applied some functions to hide the main features of the image to encrypt it.	<ul style="list-style-type: none"> - Used in frequency domain. - Uses DWT to transform. - Lossless image encryption.
[9]	Blowfish Encryption	A hybrid block-based algorithm which is based on both, image transformation techniques and Blowfish encryption & decryption algorithm. By transforming the image block then entering the result block by Blowfish algorithm.	<ul style="list-style-type: none"> - Used in frequency domain. - Lower correlation between pixels. - Higher entropy.

[10]	Rijndael Encryption	A new hybrid permutation algorithm based on both image permutation and Rijndael algorithm.	<ul style="list-style-type: none"> - Permutation algorithm. - Lower correlation between pixels. - Higher entropy.
[15]	Chaos-Based Encryption	A new algorithm using both stream cipher and chaotic maps, by defining two chaotic maps and a 104-bit size secret key for image encryption. To decorrelate the relation between pixels.	<ul style="list-style-type: none"> - Chaos based. - Lower correlation between original and encrypted image. - Different key for each pixel.
[16]	Chaos-Based and Genetic Algorithm	A technique which uses the chaotic maps and genetic algorithm for digital image encryption.	<ul style="list-style-type: none"> - Chaos based. - Genetic Algorithm.
[11]	Advanced Encryption Standard	They modified the traditional AES algorithm by adding a key stream generator to the Advanced Encryption Standard in order to ensure encryption performance improvement by reducing the entropy	<ul style="list-style-type: none"> - Modified AES. - Superiority of the modified technique over traditional encryption techniques.
[24]	Hill Cipher	They proposed a method to produce a self-invertible matrix for Hill Cipher technique.	<ul style="list-style-type: none"> - Hill Cipher. - Lower Complexity of finding matrix inverse
[25]	Hill Cipher	They proposed a new modified advanced Hill encryption algorithm (AdvHill), which uses a mandatory key matrix. The proposed scheme avoids the problems of homogeneous background images during the encryption process.	<ul style="list-style-type: none"> - Hill Cipher. - More reliable than original Hill cipher
[26]	Permutation Based Encryption	A new digital image encryption technique which is based on a randomly generated pixel permutation, with emphasize to preserve the image quality	<ul style="list-style-type: none"> - Permutation based. - Produces secret shares.
[17]	Chaos-Based Encryption	They proposed a new encryption technique by using four different chaotic maps, then they add some noise on the encrypted image then they decrypt it back.	<ul style="list-style-type: none"> - Chaos based. - Noise applied. - Less noise effect on the original image.
[28]	Chaos-Based Encryption	They proposed an encryption technique using two different chaotic maps together: The Rössler and Lorenz chaotic maps.	<ul style="list-style-type: none"> - Chaos based. - Stronger security algorithm.

[27]	Differential Evolution	They proposed a new technique for image encryption by using the Differential Evolution approach. The work depends on deploying the discrete Fourier transform, then for encryption process the differential evolution operations will be deployed.	<ul style="list-style-type: none"> - Diff. Evolution. - Fully distorted encrypted image. - Hardness of breaking the algorithm.
[12]	Advanced Encryption Standard	They introduced a standard to study the spread S-boxes and analyse their behaviour to decide if they are suitable for image encryption applications or not.	<ul style="list-style-type: none"> - Proved the suitability of an S-box to image encryption applications.
[19]	Chaos-Based Encryption	They proposed an algorithm which is based on pixel mixing, where chaos randomness is used to scramble pixel positions.	<ul style="list-style-type: none"> - Chaos based. - Uses two maps. - Selects the best map for encryption.
[13]	Advanced Encryption Standard	They proposed a new digital image encryption technique based on the rotation of the Magic Cube faces. Then the mixed image is entered to AES algorithm.	<ul style="list-style-type: none"> - AES. - Better confidentiality than other encryption methods.
[20]	Chaos-Based Encryption	He proposed a new image encryption technique based on 4D chaotic system. In order to enlarge the key space, three control parameters are added to the 4D chaotic system parameters.	<ul style="list-style-type: none"> - Chaos based. - Better efficiency. - Fast performance. - High Security.
[28]	RC6 Encryption	They studied the efficiency of RC6 encryption on digital images. Also, they proved that RC6 encryption is highly reliable and secure for real time digital image encryption.	<ul style="list-style-type: none"> - RC6. - High reliable for real image time encryption.
[21]	Chaos-Based Encryption	He proposed a novel digital image encryption technique based on chaos theory. The proposed technique relies on finding a combination between the two adjacent pixels in order to create linear independence relationships in the same row.	<ul style="list-style-type: none"> - Chaos based. - The hardness of breaking against different attacks. - Sensitive to the initial conditions.
[22]	Chaos-Based Encryption	They proposed a digital image encryption technique based on chaotic sequences. The Encryption and decryption keys are acquired by 1-D Logistic map which generates the secret key for the input of the nonlinear function.	<ul style="list-style-type: none"> - Chaos based. - Decrypt the transmitted image correctly and reliably.

[14]	Advanced Encryption Standard	They proposed a modified Advanced Encryption Standard (MAES) to be used for digital image encryption. The original AES is modified by adjusting the Shift-Row phase.	<ul style="list-style-type: none"> - AES. - Efficient and reliable technique in comparison with original AES.
[23]	Chaos-Based Encryption	Proposed a digital image encryption technique. Based on hierarchical combination of three chaotic maps the author proposed an n-ary key stream generator.	<ul style="list-style-type: none"> - Chaos based. - Secure against different types of attacks.

Chapter 3

The Proposed Techniques and Experimental Implementation

In This chapter, a brief overview of WSN is shown, then more details are described about the proposed encryption and decryption techniques using both DCT and DWT using their flowcharts. After that, we show the different scenarios and network topologies for our experiments. Finally, the performance metrics for experimental testing are explained.

3.1 Wireless Sensor Networks

The Wireless Sensor Network [29] is defined as a set of wireless nodes that have a wireless transceiver for sending and receiving the data which is collected either by the integrated sensors or sent by a user. A WSN consists of many nodes, each node is connected to all other nodes in its transmission range, all nodes together form the WSN. In WSN the nodes are classified into three categories: Sensor, Intermediate/Relay and Sink/Base Station nodes [29]. The sensor node contains different types of sensors, such as light, temperature and humidity sensors which are used to collect data from the environment and send it to the sink nodes for further processing or to be sent to another sink nodes or networks. The intermediate nodes forward the data packets from one node to another in its transmission range until it reaches from the sender to receiver or the base station node. Knowing that any node can act either as sensor or intermediate node at the same time, depending on the current situation[29].

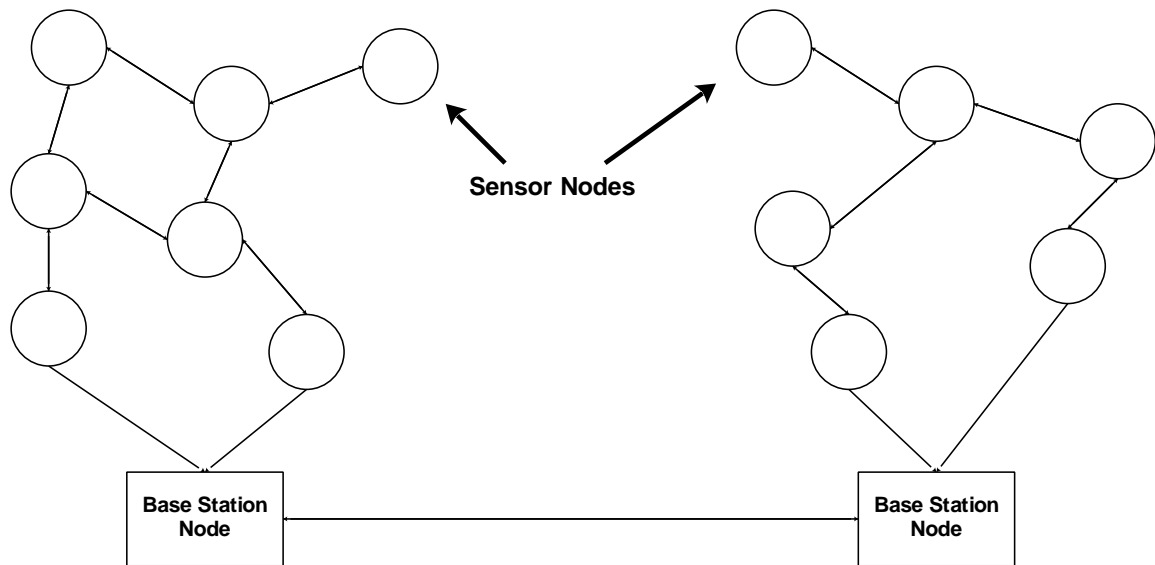


Figure 3.1: The Structure of Wireless Sensor Networks

Figure 3.1 Shows a typical structure of two different wireless sensor networks. The main components of any sensor node are: The power supply which is used to supply the node with the required power, and it's the most important part in terms of power efficiency. The sensor module is used to sense the environment and collect the data. Many types of sensor can be used in WSN such as temperature, humidity and light sensors. The microcontroller which is used to process, save and manipulate the data collected by the sensor module. The last component is the radio module which is used to transmit and receive the data from and to the other sensor nodes or to the base station [29].

When a wireless sensor network is created, the deployment of sensor nodes is done randomly based on the network and application requirements, then, the nodes start sensing and collecting the data based on a specific triggers or actions such as a temperature threshold for firefighting systems, or even when a user sends a data from source node to destination using the WSN...etc. Then the collected data will be transferred throw-out the network until it reaches the sink node for further processing or to be sent to another WSN

as we can see in Figure 3.1, or even it might be sent through the internet to be processed in different location [29].

3.2 The Main Components of WSN

As shown in Figure 3.1, the main components of the wireless sensor network are the sensor and the base station nodes, the operations of these nodes are sensing, processing, saving and sending the data.

3.2.1 The Functional Units of The Sensor Node

Each sensor node consists of four modules: Sensors modules, Power Supply, Microcontroller and Memory and at last the Radio Transceiver. Each module has its own specific job and all together builds the sensor node.

Sensor Module

Sensor is a device used to measure the change in the physical condition of an area of interest and gives response to that change. Sensors sense the environment, collect data and convert it to physical data (voltage or current) before sending it for further processing. There are different types of sensors that can be used according to the required operation. The size of sensors and their energy consumptions are important factors that need to be considered before selecting the sensors.

Power Supply

A sensor node has a power unit to provide power to all other units because energy is required for computation and data transfer. The power is mostly consumed in computation and transmission. Of the two the transmission entity consumes more power. Mostly the sensor nodes are battery operated.

Microcontroller and Memory

This microcontroller has several other sub-components such as processing unit, memory, converters and universal asynchronous receive and transmit (UART). This processing unit is responsible for handling data gathering and incoming and outgoing transmission.

The memory unit of sensor node stores the program and code. read-only memory (ROM) is used to store the data packets. While to store the program code, electrically erasable programmable read-only memory (EEPROM) or Flash memory is used.

Radio Transceiver

The networking can be obtained by the sensor nodes using radio signals or optical communication. The radio units in the sensor nodes use electromagnetic spectrum to transfer the data to required destination.

3.2.2 The Base station / Sink Node

It is an interface between sensor network and management center or another network as in Figure 3.1. In a sensor network, there can be single or multiple base station(s). The multiple base stations can perform better and decrease network delay. Base station can be stationary or dynamic (Movable).

3.3 Applications of Wireless Sensor Networks

The applications of WSN can be divided into three main categories: monitoring, tracking and data collection and transmission applications. Figure 3.2 shows the three main categories for WSN applications.

Monitoring Applications can also be divided into four sub- categories: Areas and locations monitoring, Environmental and earth sensing, Industrial monitoring and Health care

monitoring. In terms of Tracking applications, the WSN can be used for objects tracking such as cars in the roads or enemies in a battle or even animals in a forest...etc. Also, an important use of WSN is for data collection and transmission, such as collecting the temperature and humidity in different areas for time interval for a specific study, or even to check the amount of lighting in different places in different times, and, capturing images and sounds from different places and send them to another WSN or data centers for further processing.

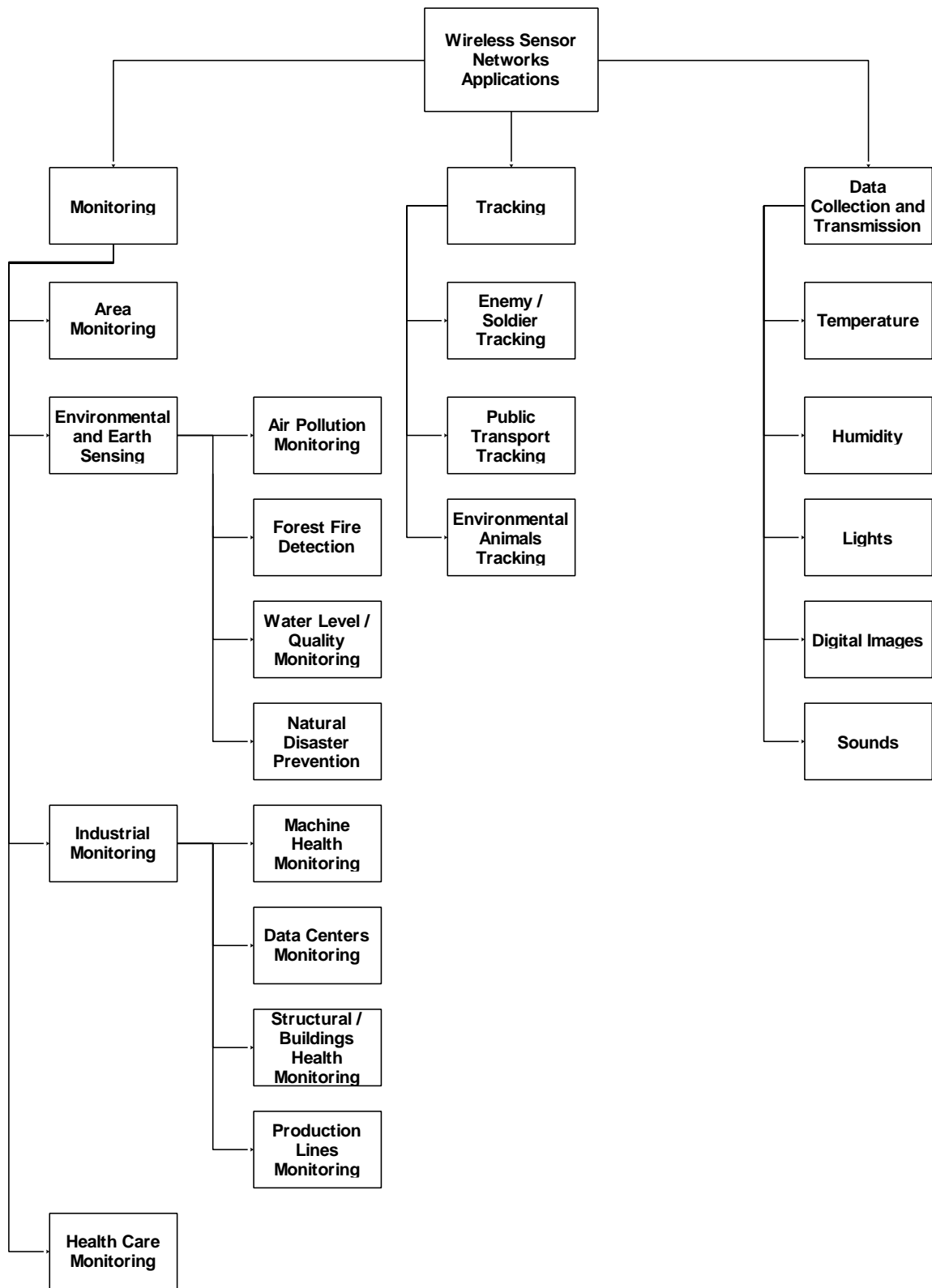


Figure 3.2: Wireless Sensor Networks Applications

3.4 Challenges in Designing a WSN

When designing a wireless sensor network for a specific application there must be many challenges must be considered in terms of Energy Efficiency, Radio Interference, Security and Data Management. Next, we will show more details about those challenges:

Energy Efficiency

The energy efficiency can be improved in several ways such as optimizing software and hardware design, which reduces the energy consumption and makes WSN energy efficient. Another method is to optimize the power management at hardware and network levels.

Radio Interference

The performance of WSNs can be severely affected by the other wireless system working in the same frequency spectrum and in the same area. Interference avoidance algorithms can be used to overcome the limitations of WSN such as slow computation.

Security

The wireless nature of WSN can cause some security issues, which are unavoidable. The data distributions must be safeguarded from unauthorized access.

Data Management

When a huge volume of data is sent over the network, the cost of such transmission in terms of energy consumption is very high. So, the data compression and aggregations techniques can be used to compress or to arrange the data. Robust techniques can be used to reduce the amount of data while utilizing the data processing time.

3.5 The Proposed Technique Using DCT

In this part, a digital image encryption technique using the DCT is proposed to be used for WSN. As shown in Figure 3.3, both encryption and decryption algorithms were applied on N by N pixel image, by dividing the image into blocks of size m by n .

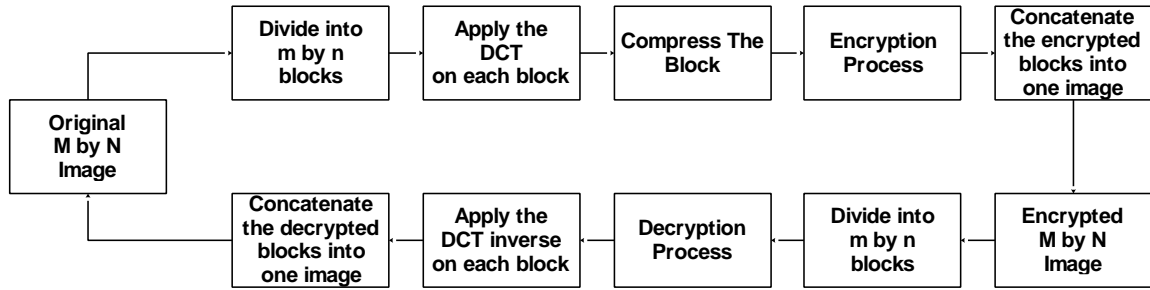


Figure 3.3: General Overview of The Encryption and Decryption using DCT

At the sender, the encryption technique uses the DCT to transform and compress the original/plain image block from pixel to frequency domain. Then, encryption process is done.

At the receiver side, the decryption algorithm is done using the DCT inverse on the encrypted image blocks to retrieve the image back to its original structure by reversing the encryption process steps. In both encryption and decryption processes, a key matrix is used of m by n block size, this key matrix is applied on each block of the image separately.

At the Sender Side

The key matrix, is an auto generated random matrix of size m by n , the values of this matrix are between 1 and 256. Each communication line between each two nodes needs a different key matrix, for example let's suppose that we have three different buildings which needs to communicate with each other as shown in Figure 3.4, each building/node of those must have two keys; one for each other building. The keys are predefined or shared for each sender/receiver node to be used any time later for encrypting and decrypting images. Only the desired destination can decrypt the image because its already encrypted using the pre-shared key between the sender and desired receiver.

The details of the encryption algorithm are shown in Figure 3.5. The first step, that we have an image of size N by N , in this step we used a standard benchmark images for experimental uses. The image is divided into equal size of blocks of size m by n to get several blocks in pixel domain and then we do the encryption processes.

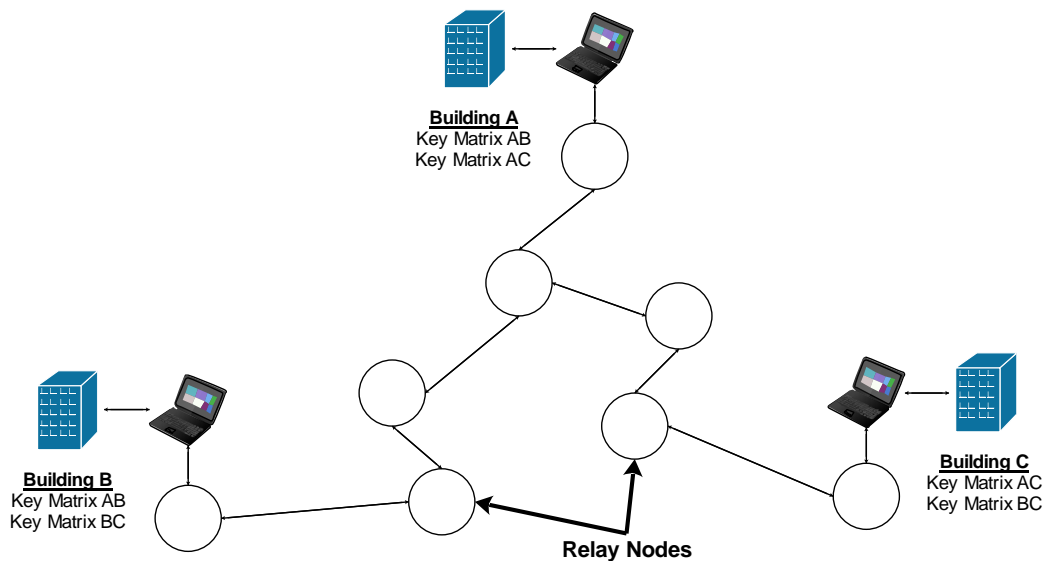


Figure 3.4: The Distribution of Key Matrices

Now, for all original blocks in pixel domain we do the following:

- Apply the DCT to transform the block from pixel to frequency domain. Now, we have the original image block in frequencies.
- Compress The Image Block
- For the block, we rotate the rows values row by row. Each row is rotated X times. The number of rotations X_i for each row is calculated by getting the first column of the key matrix W , then we will rotate the i^{th} row using equation 3.1:

$$X_i = ((m*n) + W(i)) \% m \quad (3.1)$$

Where ‘ m ’ and ‘ n ’ are the dimensions of the block, ‘ W ’ is the first column of the key matrix.

- Then, the values of the blocks are scattered to change the original values to another decorrelated values, the scattering is done by expanding each value of the image block $I(i, j)$ using equation 3.2, then dividing each value of the block $I(i, j)$ by the corresponding value of the key matrix $K(i, j)$ divided by ‘ m by n ’ using equation 3.3.

$$I(i, j) = I(i, j) / 0.001 \quad (3.2)$$

$$I(i, j) = I(i, j) / (K(i, j) / (m*n)) \quad (3.3)$$

Where ‘ I ’ is the image block, ‘ K ’ is the key matrix, ‘ m ’ and ‘ n ’ are the dimensions of the block.

- The next step is the block values transpose. Which means that we rearrange the rows to be columns and rearrange the columns to be rows.
- After that, we perform the block values shuffling around the diagonal, which means that we swap the values of the block in symmetrical order around the main diagonal.
- At this point we got the encrypted image block. While it's still in the frequency domain.
- We repeat the previous steps for all the blocks until we got all the encrypted image blocks.
- At this stage, we concatenate all the encrypted image blocks together.
- Now, we have the needed encrypted and compressed N by N image in frequency domain.
- The last step, is that we send the encrypted image from the sender node to the designated or destination node throughout the WSN.

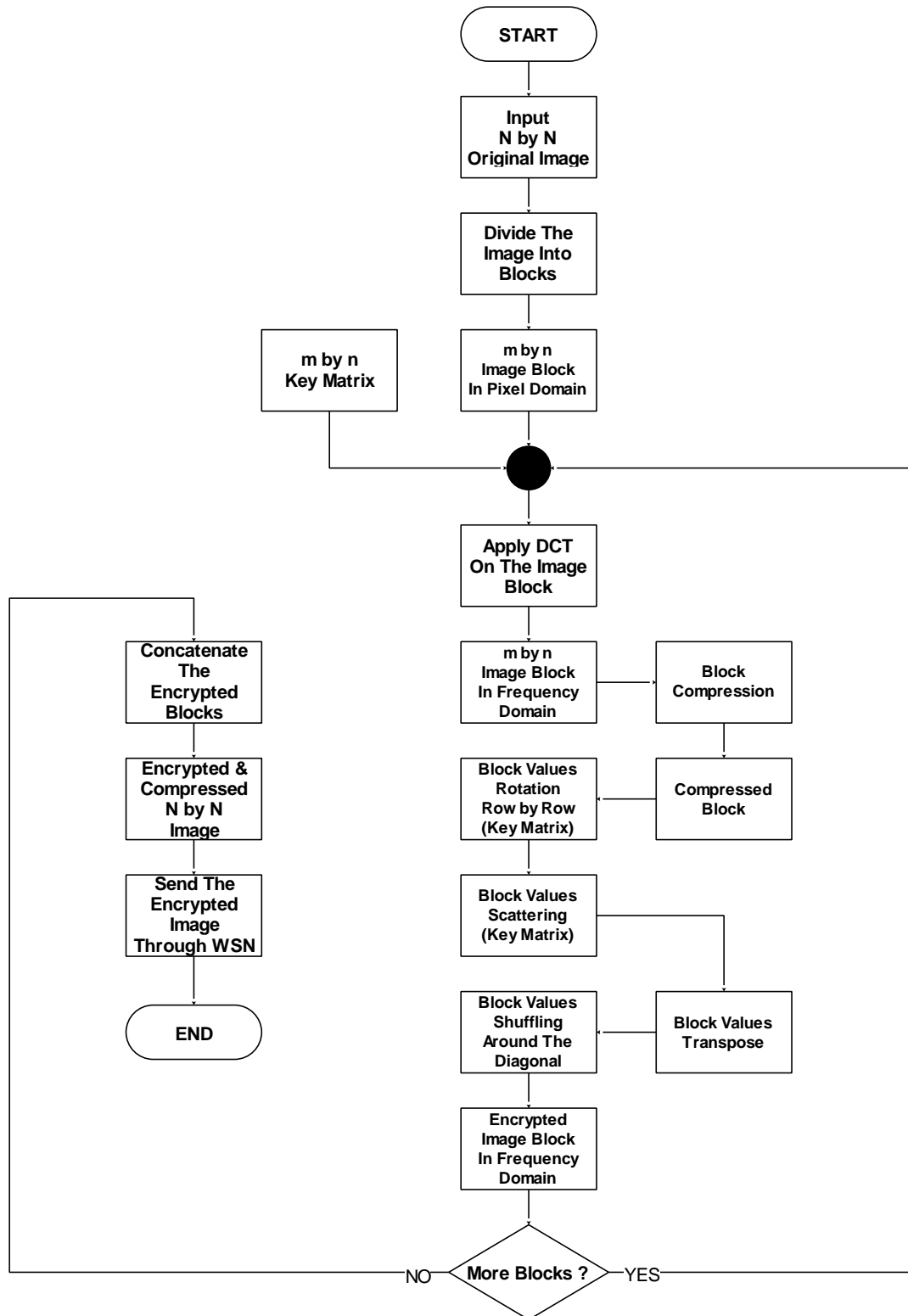


Figure 3.5: DCT Encryption Algorithm Flowchart

Pseudo Code:

I: Original NxN Image

I: Apply DCT Transform

I2: Compressed I

For each block of size MxN of I2

Do:

- **Rotate the block values using the key matrix**
- **Block values Scattering**
- **Block values Transpose**
- **Block values Shuffling**

End

I3: Compressed and Encrypted Image in Frequency Domain

At the Receiver Side

The details of the decryption algorithm are shown in Figure 3.6. The first step, that we receive the encrypted image throughout the WSN. We download the received image from the sensor node to get an encrypted image of size N by N , in this step we used a standard benchmark images for experimental uses. The image now is divided into equal size of blocks of size m by n to get several blocks in frequency domain and then do the decryption processes.

As mentioned in Figure 3.4, the key matrix is predefined and shared among all senders and receivers in the WSN. So, in the decryption algorithm we must use the same key matrix that been used in the encryption algorithm.

Now, for all encrypted blocks in frequency domain we do the following:

- We reverse the block values shuffling around the diagonal, which means that we swap the values of the block in symmetrical order again around the main diagonal to return them back to the original order.
- The next step is the block values transpose. Which means that we rearrange the rows to be columns and rearrange the columns to be rows again.
- Then, the values of the blocks are retrieved back to their original form after they being scattered in the encryption algorithm. The retrieving is done by multiplying each value of the block $I(i, j)$ by the corresponding value of the key matrix $K(i, j)$ divided by ' m by n ' using equation 3.4. Then, each value of the image block $I(i, j)$ is collapsed back using equation 3.5

$$I(i, j) = I(i, j) * (K(i, j) / (m*n)) \quad (3.4)$$

$$I(i, j) = I(i, j) * 0.001 \quad (3.5)$$

Where I is the encrypted image block, K is the key matrix, m and n are the block dimensions.

- Then, we rotate back the rows values row by row. Each row is rotated X times. The number of rotations X_i for each row is calculated by getting the first column of the key matrix W , then we will rotate the i^{th} row using equation 3.6:

$$X_i = m - (((m*n) + W(i)) \% m) \quad (3.6)$$

Where ' m ' and ' n ' are the dimensions of the block, ' W ' is the first column of the key matrix.

- At this point we got the decrypted image block. While it's still in the frequency domain.
- We apply the DCT inverse to transform the image block from frequency to pixel domain.
- Now, we have the decrypted and compressed image block in the pixel domain.
- We repeat the previous steps for all the blocks until we got all the decrypted image blocks.
- At this stage, we concatenate all the decrypted image blocks together.

- Now, we have the needed compressed and decrypted image.

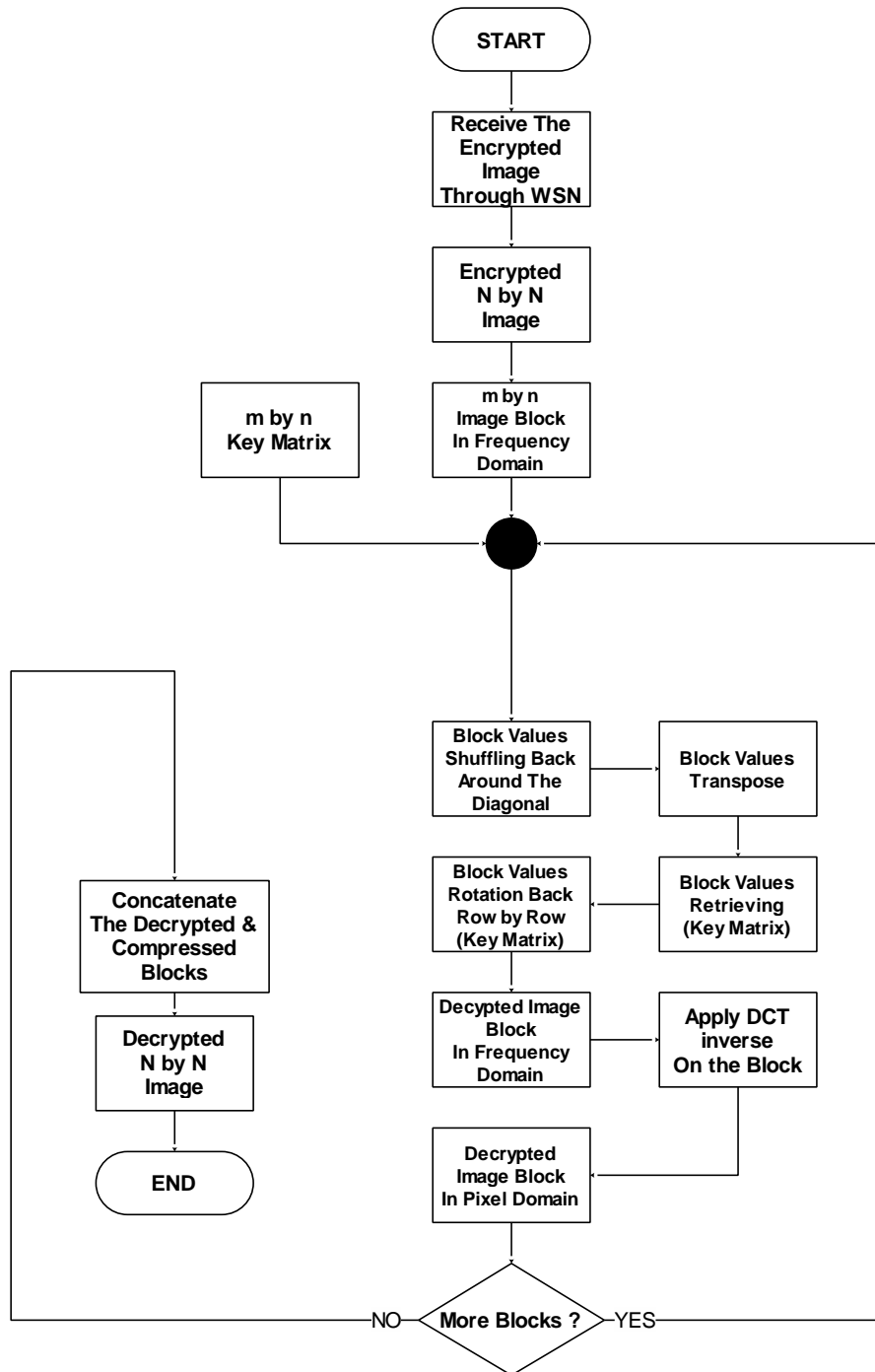


Figure 3.6: DCT Decryption Algorithm Flowchart

Pseudo Code:

I: Encrypted NxN Image

For each block of size MxN of I

Do:

- **Block values Shuffling**
- **Block values Transpose**
- **Block values Scattering**
- **Rotate the block values using the key matrix**

End

I2: Decrypted NxN image in Frequency Domain

I3: Apply DCT inverse to get the Decrypted image in pixel domain

3.6 The Proposed Technique Using DWT

In this part, a digital image encryption technique using the DWT is developed to be used for WSN. As shown in Figure 3.7, both encryption and decryption algorithms were applied on N by N pixel image.

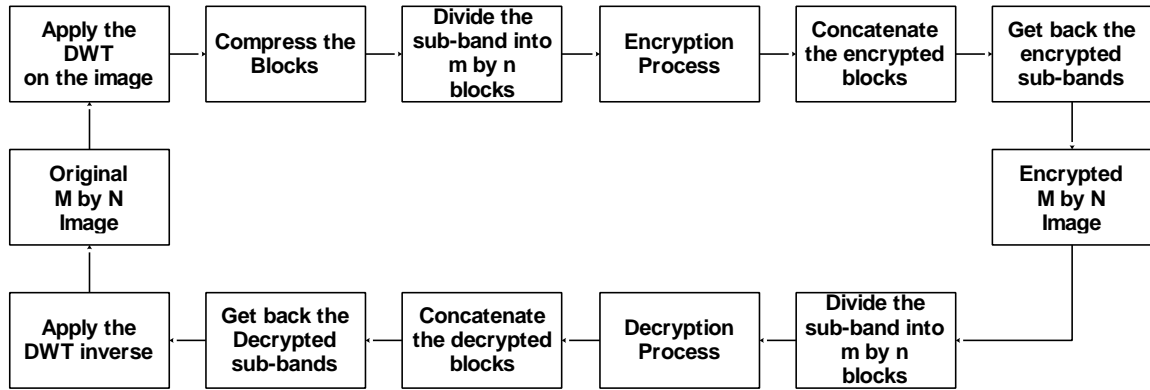


Figure 3.7: General Overview of The Encryption and Decryption using DWT

At the sender, the encryption technique uses the DW transform to transform the original/plain image block from pixel to frequency domain, because of the nature of DWT which is applied on the entire image not on the blocks, we applied the DWT on the image to get the four sub-bands HH, HL, LH and LL, and then we divided each sub-band into m by n blocks, after that, the encryption process is done on the image in the frequency domain to get the encrypted image. At the receiver side, the decryption algorithm applies the DWT inverse on the encrypted image to restore it back to its original form by reversing back the encryption process. In both encryption and decryption, a key matrix is used of m by n block size, this key matrix is applied on each block of the image separately.

At the Sender Side

The details of the encryption algorithm are shown in Figure 3.8. The first step, that we have an image of size N by N , in this step we used a standard benchmark images for experimental uses.

We will apply the first level of the Discrete Wavelet Transform DWT on the N by N image to get the four sub-bands LL, LH, HL and HH. Then, we do a compression. After that, we will apply the second level of DWT on the $N/2$ by $N/2$ LL, to get the four sub-bands LL_2 , LH_2 , HL_2 and HH_2 .

After that, we will divide each one of LL_2 , LH_2 , HL_2 and HH_2 into m by n blocks, the sub-bands now are divided into equal size of blocks in the frequency domain. and then we do the encryption processes.

The key matrix, is an auto generated random matrix of size m by n , the values of this matrix are between 1 and 256. Each communication line between each two nodes needs a different key matrix, for example let's suppose that we have three different buildings which needs to communicate with each other as shown in Figure 3.4, each building/node of those must have two keys; one for each other building. The keys are predefined or shared for each sender/receiver node to be used any time later for encrypting and decrypting images. Only the desired destination can decrypt the image because its already encrypted using the pre-shared key between the sender and desired receiver.

Now, for each original block of the four sub-bands in the frequency domain we do the following:

- For the block, we rotate the rows values row by row. Each row is rotated X times. The number of rotations X_i for each row is calculated by getting the first column of the key matrix W , then we will rotate the i^{th} row using equation 3.1 which is mentioned in the previous section.
- Then, the values of the blocks are scattered to change the original values to another decorrelated values, the scattering is done by dividing each value of the block $I(i, j)$ by the corresponding value of the key matrix $K(i, j)$ divided by 'm by n' using equation 3.3 which is mentioned in the previous section.
- The next step is the block values transpose. Which means that we rearrange the rows to be columns and rearrange the columns to be rows.
- After that, we perform the block values shuffling around the diagonal, which means that we swap the values of the block in symmetrical order around the main diagonal.
- At this point we got the encrypted image block. While it's still in the frequency domain.
- We repeat the previous steps for all the blocks until we got all the encrypted image blocks in the frequency domain.
- At this stage, we concatenate all the encrypted image blocks together.
- Now, we have the needed encrypted $N/2$ by $N/2$ LL_2 , LH_2 , HL_2 and HH_2 .

- For each value (i,j) of the four sub-bands LL_2 , LH_2 , HL_2 and HH_2 , we will swap the values of LL_2 with HH_2 , and we will swap the values of LH_2 with HL_2 .
- We will apply the DWT inverse on the four sub-bands LL_2 , LH_2 , HL_2 and HH_2 to get the encrypted LL .
- For each value (i,j) of the four sub-bands LL , LH , HL and HH , we will swap the values of LL with HH , and we will swap the values of LH with HL .
- Now, we have the needed encrypted and compressed N by N image in frequency domain.
- The last step, is that we send the encrypted image from the sender node to the designated or destination node throughout the WSN.

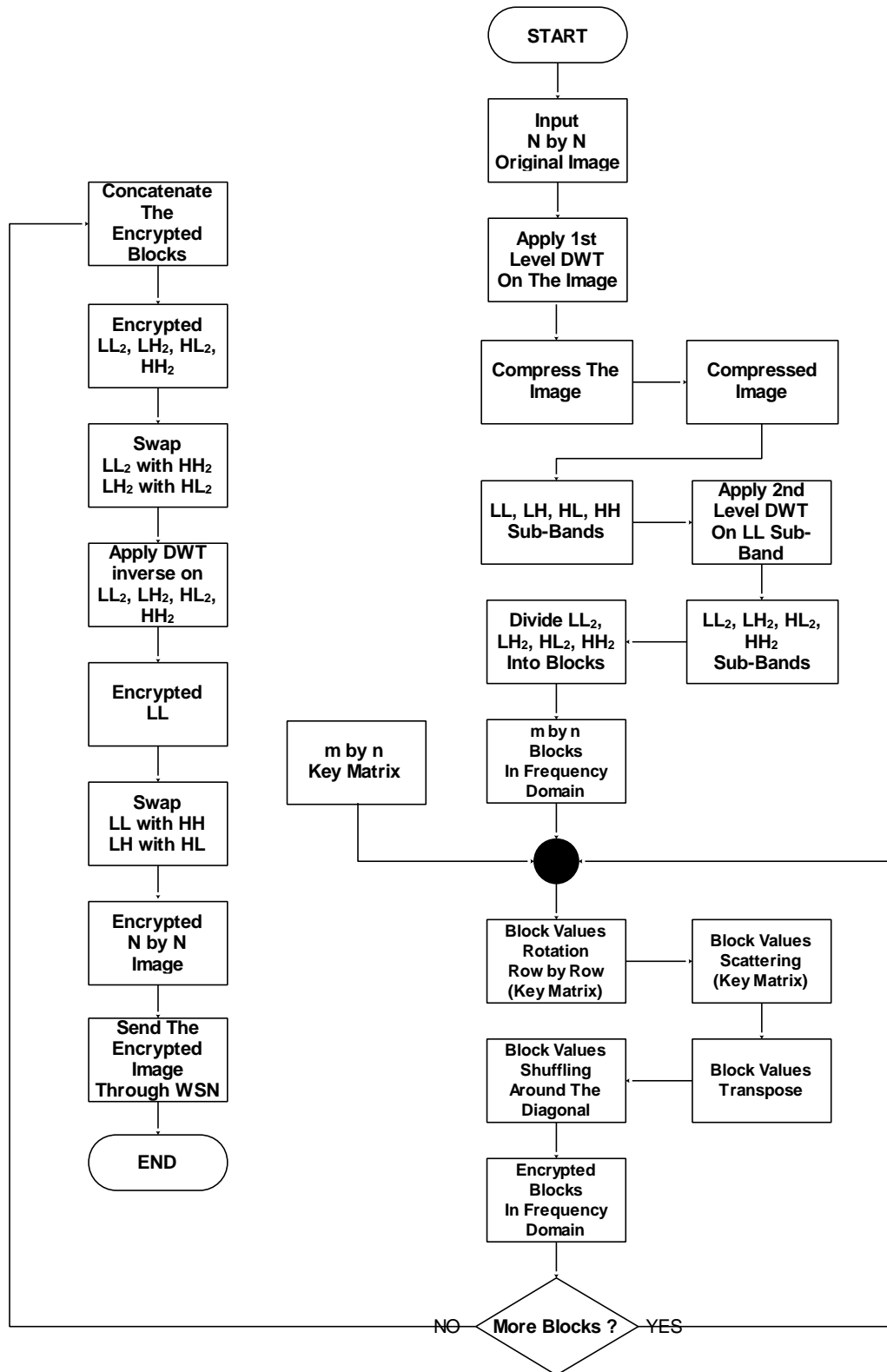


Figure 3.8: DWT Encryption Algorithm Flowchart

Pseudo Code:

I: Original NxN Image

I: Apply 1st level DWT Transform

I2: Compressed I

I3: Apply 2nd level DWT Transform

For each block of size MxN of I3

Do:

- **Rotate the block values using the key matrix**
- **Block values Scattering**
- **Block values Transpose**
- **Block values Shuffling**

End

Swap between LL₂ and HH₂

Swap between LH₂ and HL₂

Apply 2nd level DWT inverse

Swap between LL and HH

Swap between LH and HL

I3: Compressed and Encrypted Image in Frequency Domain

At the Receiver Side

The details of the decryption algorithm are shown in Figure 3.9. The first step, that we receive the encrypted image throughout the WSN. We download the received image from the sensor node to get an encrypted image of size N by N , in this step we used a standard benchmark images for experimental uses.

For each value (i,j) of the four sub-bands LL, LH, HL and HH, we will swap the values of LL with HH, and we will swap the values of LH with HL, in order to get the encrypted sub-band LL.

Then, we will apply the second level of DWT on the encrypted $N/2$ by $N/2$ LL, to get the encrypted four sub-bands LL_2 , LH_2 , HL_2 and HH_2 .

After that, we will divide each one of LL_2 , LH_2 , HL_2 and HH_2 into m by n blocks, the sub-bands now are divided into equal size of blocks in the frequency domain.

As mentioned in Figure 3.4, the key matrix is predefined and shared among all senders and receivers in the WSN. So, in the decryption algorithm we must use the same key matrix that been used in the encryption algorithm.

Now, for each encrypted block of the four sub-bands in the frequency domain we do the following:

- We perform the block values shuffling around the diagonal, which means that we swap the values of the block in symmetrical order around the main diagonal.

- The next step is the block values transpose. Which means that we rearrange the rows to be columns and rearrange the columns to be rows.
- Then, the values of the blocks are retrieved back to their original form after they being scattered in the encryption algorithm. The retrieving is done by multiplying each value of the block $I(i, j)$ by the corresponding value of the key matrix $K(i, j)$ divided by 'm by n' using equation 3.4 which is mentioned in the previous section.
- Then, we rotate back the rows values row by row. Each row is rotated X times. The number of rotations X_i for each row is calculated by getting the first column of the key matrix W , then we will rotate the i^{th} row using equation 3.6 which is mentioned in the previous section.
- At this point we got the decrypted image block. While it's still in the frequency domain.
- We repeat the previous steps for all the blocks until we got all the decrypted image blocks in the frequency domain.
- At this stage, we concatenate all the decrypted image blocks together.
- Now, we have the needed decrypted $N/2$ by $N/2$ LL_2 , LH_2 , HL_2 and HH_2 .
- For each value (i, j) of the four sub-bands LL_2 , LH_2 , HL_2 and HH_2 , we will swap the values of LL_2 with HH_2 , and we will swap the values of LH_2 with HL_2 .
- We will apply the DWT inverse on the four sub-bands LL_2 , LH_2 , HL_2 and HH_2 to get the decrypted LL .

- We will apply the DWT inverse on the four sub-bands LL, LH, HL and HH to get the decrypted image.
- Now, we have the needed decrypted and compressed N by N image in pixel domain.

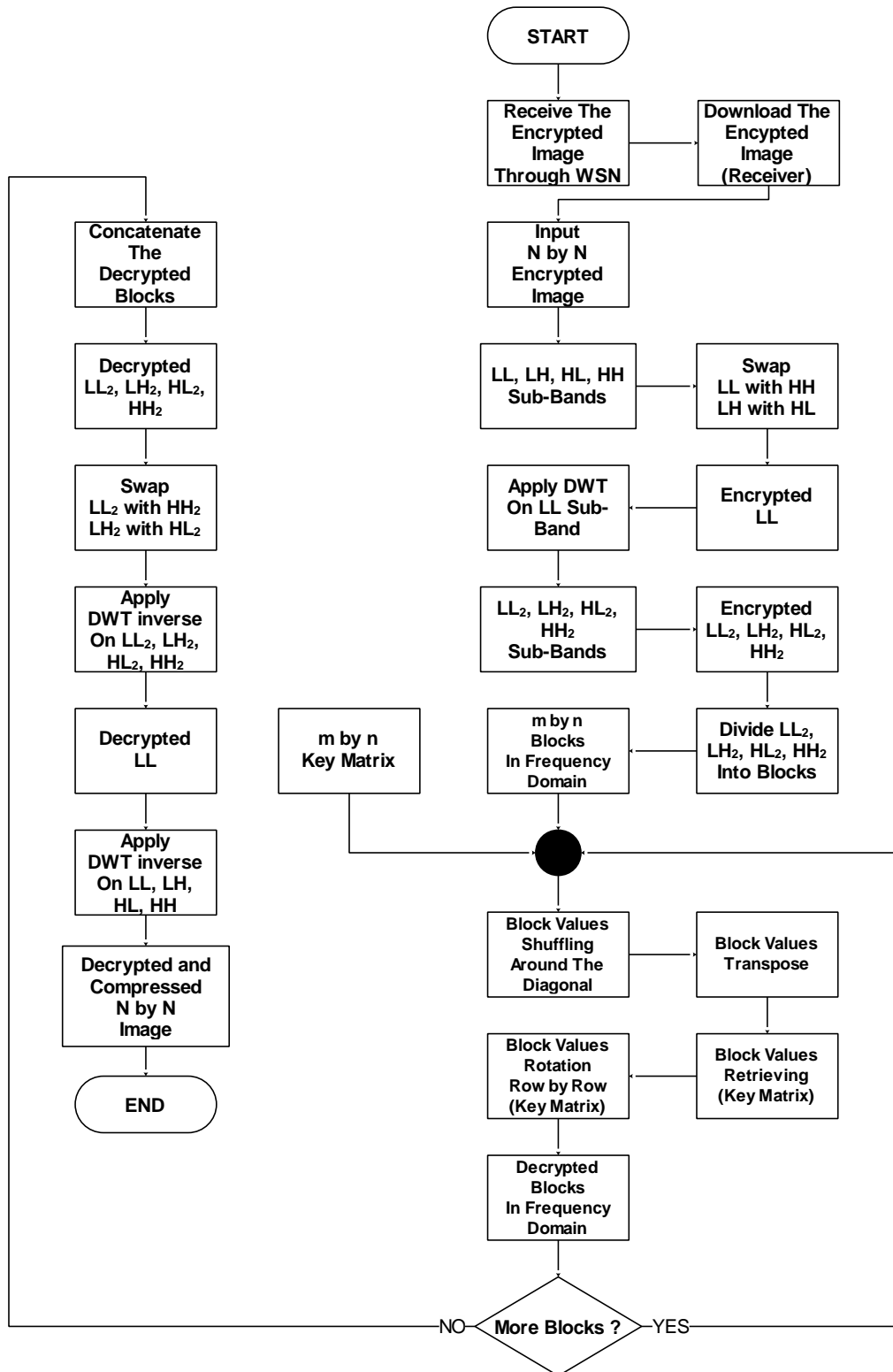


Figure 3.9: DWT Decryption Algorithm Flowchart

Pseudo Code:

I: Encrypted NxN Image

Swap between LL and HH

Swap between LH and HL

I2: Apply 2nd level DWT Transform

For each block of size MxN of I2

Do:

- **Block values Shuffling**
- **Block values Transpose**
- **Block values Scattering**
- **Rotate the block values using the key matrix**

End

Swap between LL₂ and HH₂

Swap between LH₂ and HL₂

I3: Apply 2nd level DWT inverse

I3: Apply 1st level DWT inverse get the Decrypted image in pixel domain

3.7 Experimental Implementation

The implementation of the proposed idea has been done using the tools and hardware listed in Table 3.1. The implementation of the proposed encryption and decryption algorithms for both DCT and DWT were done on Matlab R2015a, we choose Matlab because its ability to deal with digital images as matrices and then do the needed mathematical processes. As WSN, we used the Instant Contiki to program our network and for Contiki we used the Cooja Simulator to simulate our different experimental networks scenarios. The sensor motes hardware that been used in all experimental scenarios were the sky mote.

Table 3.1: Tools and Software Used for Implementation:

Software / Hardware	Information
Implementation of the DCT and DWT for Encryption and Decryption algorithms	Matlab R2015a
WSN Programming	Instant Contiki 2.7
WSN Simulator	Cooja Simulator
Sensor Mote	Tmote Sky

3.8 Experimental Test Cases of Digital Images

Digital images that is used for experimental testing of any research in the fields of image processing such as image compression, encryption, restoration and enhancement is called a standard test image. For a fair comparison between different techniques in the same field of image processing, the same standard image must be used. The type of the chosen standard image lay down on the need of the proposed thesis. For example, some researches

needs a high detailed images or patterns for its experimental testing, so, the correct type of images must be chosen.

Table 3.2: Test Cases Parameters:

Parameters	Information
Encryption Techniques	DCT, DWT
Image Resolution	256x256
Block Size	8x8, 16x16, 32x32

The USC-SIPI image database [30] is used to test the proposed algorithms. The USC-SIPI database is a set or collection for standard digital images. Mainly, it is founded to standardize and help the research in the field of image processing, image analysis and machine vision. First USC-SIPI database edition was founded and distributed in 1977, and by that time, many standard images have been added. Based on the character of each set of standard images, the database is classified into volumes. Each volume contains of different sizes of standard images such as 256 by 256 pixels, 512 by 512 pixels and 1024 by 1024 pixels. All standard images are 1 byte per pixel for grayscale images, 3 bytes per pixel for colored images. Figures 3.10, 3.11 and 3.12 respectively shows the images that used as test cases for our experimental testing. Table 3.2 shows the parameters we used for the experimental test cases images.



Figure 3.10: Cameraman Image

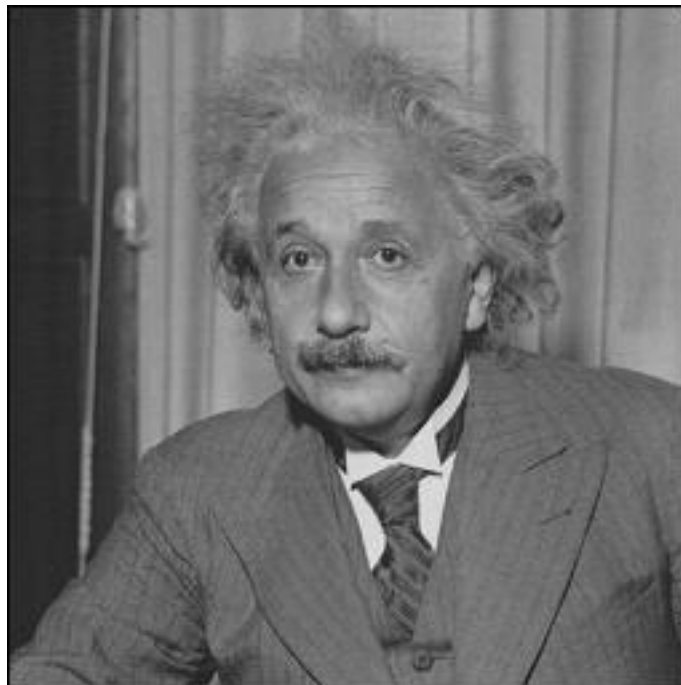


Figure 3.11: Einstein Image



Figure 3.12: Peppers Image

3.9 Experimental Networks Topologies

For experimental testing, two different types of networks topologies have been used, single-hop and multi-hop. Single-hop networks contains only two nodes: sender and receiver nodes, while the multi-hop networks contain more than two nodes: sender, receiver and intermediate nodes. In any network, the encryption was done on the sender side while the decryption is done on the receiver side. The analysis of the proposed techniques on both DCT and DWT was done using several matrices, in terms of image quality we used the SSIM, PSNR and Histogram analysis, for QoS we used the End-to-End Delay.

3.9.1 Single-Hop Network

In the single-hop scenario, we used only two sensor nodes, the sender and the receiver or destination node. The encryption is done on the sender side, while the decryption is done

on the receiver side. We used the standard IEEE 802.15.4 protocol for the communication between sensor nodes. Table 3.3 shows the parameters for the single-hop scenario.

Table 3.3: Single-Hop Scenario Parameters:

Parameter	Values
Network Size	2
Number of Intermediate Nodes	0
Sensor Motes Type	Tmote Sky
Data Rate	250 Kbps
Packet Size	128 bytes
Distance Between Nodes	20 meters

Figure 3.13 shows the simulator screen for Single-Hop simulation.

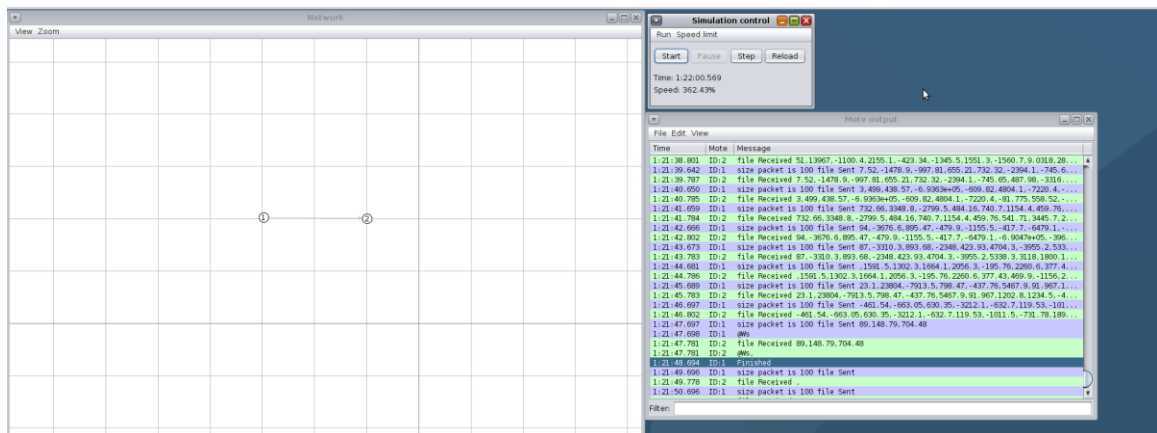


Figure 3.13: Single-Hop Simulation

3.9.2 Multi-Hop Network

In the multi-hop scenario, we used intermediate nodes. The encryption is done on the sender side, while the decryption is done on the receiver side. The intermediate nodes were used only to pass (forward) the data packets throughout the network. One scenario was experimented as multi-hop networks: three intermediate nodes. We used the standard IEEE 802.15.4 protocol for the communication between sensor nodes. Table 3.4 shows the parameters for the multi-hop scenarios.

Table 3.4: Multi-Hop Scenarios Parameters:

Parameter	Values
Network Size	15
Number of Sender Nodes	3
Number of Receiver Nodes	3
Number of Intermediate Nodes	9
Sensor Motes Type	Tmote Sky
Data Rate	250 Kbps
Packet Size	128 bytes
Distance Between Nodes	20 meters

Figure 3.14 shows the simulator screen for Multi-Hop simulation.

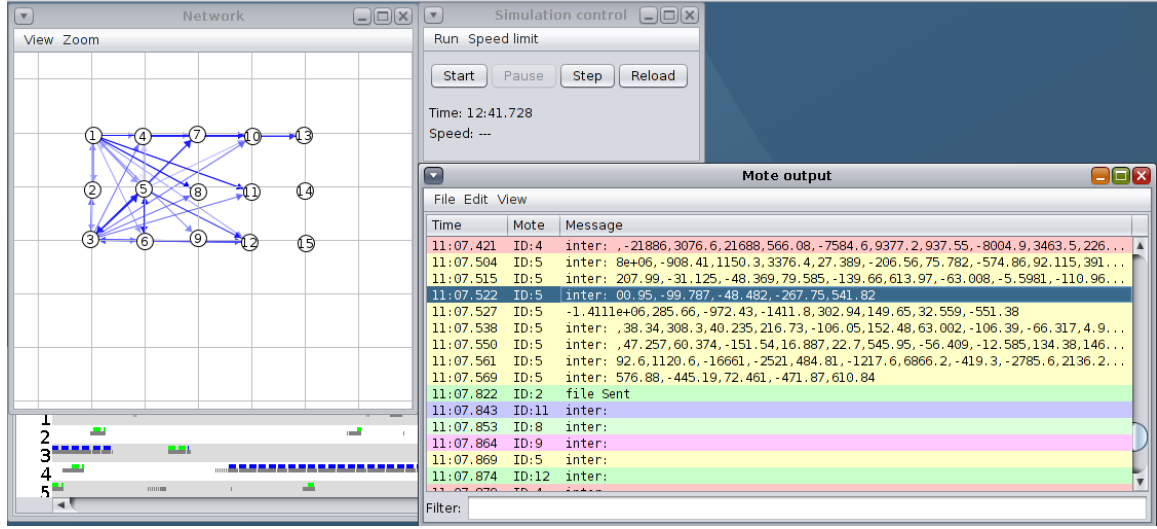


Figure 3.14: Multi-Hop Simulation

3.10 Performance Metrics for Experimental Testing

To evaluate the performance of both proposed algorithms using (DCT and DWT) we used five metrics, those metrics are categorized into two main categories: Image Quality and Network QoS. The used metrics are:

- Peak Signal to Noise Ratio (PSNR)
- Structural Similarity (SSIM)
- Histogram Analysis
- End-to-End Delay
- Energy Efficiency

3.10.1 Peak Signal to Noise Ratio (PSNR)

After any distortion, the objective quality assessments measure the quality of images. The main aim of objective evaluation is to develop quantitative measures that can expect perceived image quality. Objective quality assessments are mathematical models.

Peak Signal to Noise Ratio is most popular of the common Full reference (FR) objective quality measures (based on the difference between original and distorted image) which used to evaluate most image processing algorithms.

The peak signal to noise ratio (PSNR), is the ratio between the value of the maximum element or pixel in digital images and the value of the Mean Square Error (MSE). Equations 3.7 and 3.8 respectively shows how to calculate the MSE and PSNR.

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [X(i,j) - Y(i,j)]^2 \quad (3.7)$$

$$PSNR = 10 \log_{10} \frac{MAX^2}{MSE} \quad (3.8)$$

Where MAX is the maximum pixel value of the image, in grayscale images MAX = 255.

The PSNR is used to measure the quality of a digital image processing algorithms such as encryption and decryption algorithms. In digital images, the signal is the original image, and the encrypted/decrypted image is considered as the noise.

Typical values for the PSNR in image processing algorithms are between 30 and 50 dB not to notice any differences between the original and the processed image. For example, in image compression, the higher value is better because the compressed image is more likely

to the original. In the proposed encryption algorithms, it's better to get lower PSNR between the original and the encrypted image because both images must not be the same at all, while in the proposed decryption algorithms it's better to get higher PSNR between the original and decrypted images.

3.10.2 Structural Similarity (SSIM)

The Structural Similarity (SSIM) is an objective quality assessment for measuring the quality of two images. SSIM is a full reference assessment which means that there is a reference image as an original undistorted image and the other image is the distorted or encrypted or even compressed image. For the proposed algorithms, the similarity value between the original and encrypted images must be as small as possible for better results, while the similarity value between original and decrypted images must be as higher as possible.

To Calculate the SSIM between two images X and Y, three components must be calculated: luminance, contrast and structure of both X and Y images as in equations 3.9, 3.10 and 3.11 respectively and then combine them as in equation 3.12 [31].

$$l(X, Y) = \frac{2\mu_x\mu_y + C1}{\mu_x^2 + \mu_y^2 + C1} \quad (3.9)$$

$$c(X, Y) = \frac{2\sigma_x\sigma_y + C2}{\sigma_x^2 + \sigma_y^2 + C2} \quad (3.10)$$

$$s(X, Y) = \frac{\sigma_{xy} + C3}{\sigma_x\sigma_y + C3} \quad (3.11)$$

$$SSIM(X, Y) = l(X, Y)^\alpha \cdot c(X, Y)^\beta \cdot s(X, Y)^\gamma \quad (3.12)$$

Where μ_x and μ_y are the mean values of the images X and Y, σ_x and σ_y are the variance values of the images X and Y, σ_{xy} is the covariance of the images X and Y. C1, C2 and C3 are constants with small values to avoid the denominators when calculating the $l(X, Y)$, $c(X, Y)$ and $s(X, Y)$ from becoming zero. The constants C1, C2 and C3 are calculated using: $C1 = (K1L)^2$, $C2 = (K2L)^2$, $C3 = C2/2$, L value is the dynamic range value of the pixels which is 255 for the 8-bit grayscale images, K1 and K2 are equal to 0.01 and 0.03 respectively. α , β and γ must be greater than 0 and usually $\alpha = \beta = \gamma = 1$ [31].

3.10.3 Histogram Analysis

The histogram of encrypted images represents the density distribution of the gray level value of a given image. The histogram plot can be used as a visual inspection of how image density is distributed. In this thesis, the histogram of the original and encrypted images will be shown, images are significantly different from each other and therefore it does not afford any information useful for statistical analysis attack on encrypted images.

3.10.4 End-to-End Delay

End-to-End delay (ETE) delay is the time taken for an image to be encrypted and sent from source device until it reaches to the destination and then it being decrypted. E-E Delay consists of three components: encryption time at the sender, sending time and decryption time at the receiver or destination. The E-E Delay is computed by equation 3.13.

$$\text{End to End Delay} = (T_e + T_{send} + T_d) \quad (3.13)$$

Where,

T_e : is the needed time for image encryption at the sender.

T_{send} : is the needed time for the image to be sent through the network.

T_d : is the needed time for image decryption at the receiver.

3.10.5 Energy Efficiency

Energy efficiency is used to calculate how many images can be sent continuously for a full power battery without recharging.

Chapter 4

Experimental Results Evaluation and Discussion

In this chapter, the experimental results will be shown and discussed for both encryption techniques using DCT and DWT.

4.1 PSNR and SSIM

In this section, we will show the encrypted and decrypted test cases using the parameters shown in table 3.2. and we will show the PSNR and SSIM values of each test case and its parameters.

4.1.1 Cameraman Image using DCT Algorithm

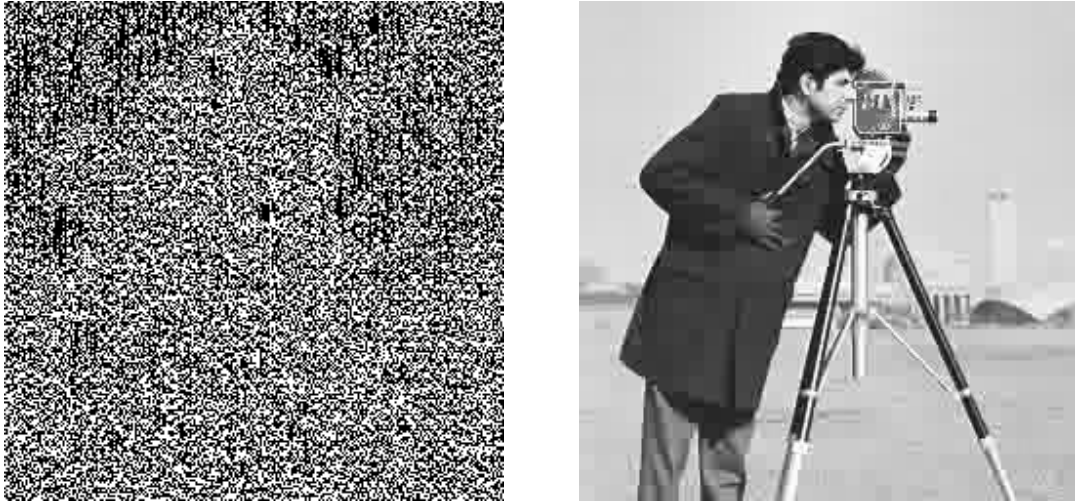


Figure 4.1: Encrypted and Decrypted Cameraman Image using DCT with 8x8 Block Size

Figure 4.1 shows both encrypted and decrypted images as results of the proposed DCT algorithm with 8 by 8 block size. Table 4.1 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.1: PSNR and SSIM results of Cameraman Image with DCT 8x8 Block Size

Block Size	PSNR	SSIM
8x8	4.4987	0.0038

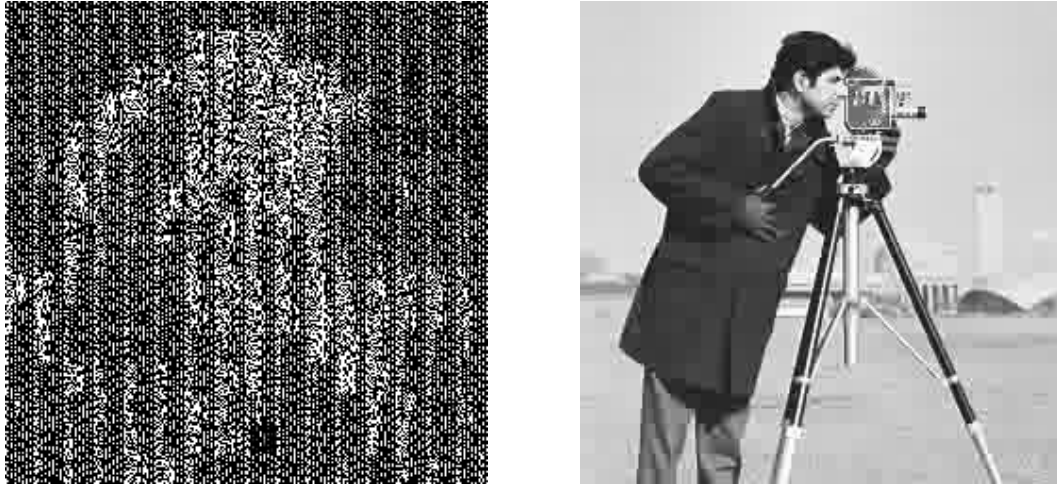


Figure 4.2: Encrypted and Decrypted Cameraman Image using DCT with 16x16 Block Size

Figure 4.2 shows both encrypted and decrypted images as results of the proposed DCT algorithm with 16 by 16 block size. Table 4.2 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.2: PSNR and SSIM results of Cameraman Image with DCT 16x16 Block Size

Block Size	PSNR	SSIM
16x16	3.8946	0.0050

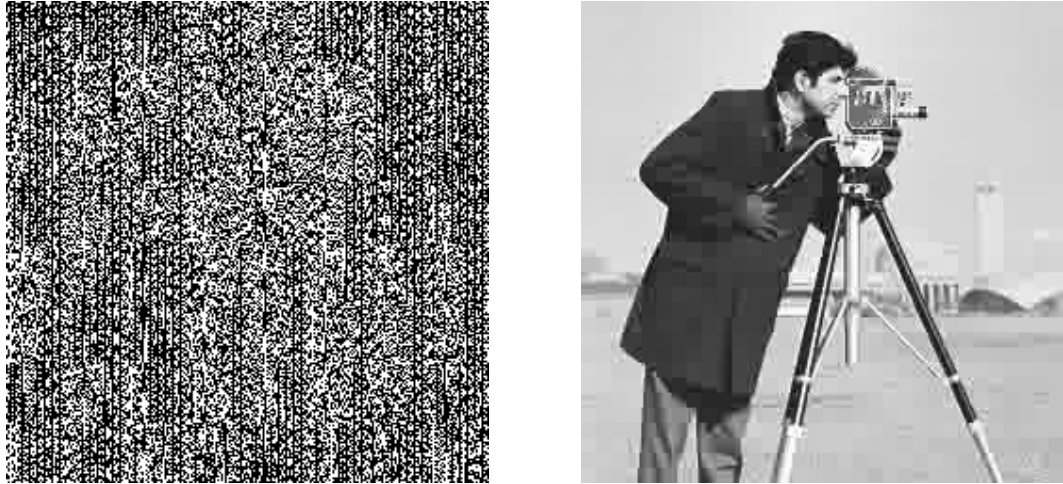


Figure 4.3: Encrypted and Decrypted Cameraman Image using DCT with 32x32 Block Size

Figure 4.3 shows both encrypted and decrypted images as results of the proposed DCT algorithm with 32 by 32 block size. Table 4.3 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.3: PSNR and SSIM results of Cameraman Image with DCT 32x32 Block Size

Block Size	PSNR	SSIM
32x32	4.3431	0.0041

4.1.2 Einstein Image using DCT Algorithm

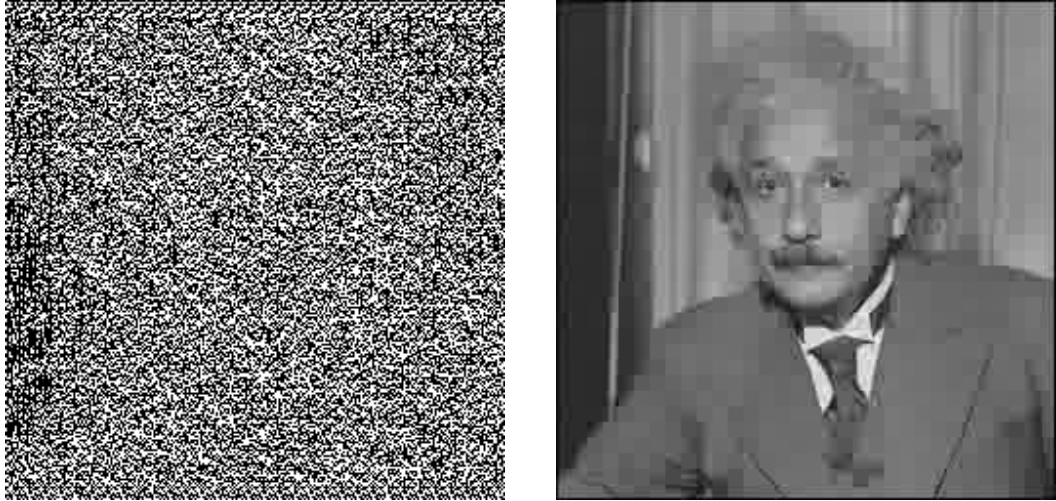


Figure 4.4: Encrypted and Decrypted Einstein Image using DCT with 8x8 Block Size

Figure 4.4 shows both encrypted and decrypted images as results of the proposed DCT algorithm with 8 by 8 block size. Table 4.4 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.4: PSNR and SSIM results of Einstein Image with DCT 8x8 Block Size

Block Size	PSNR	SSIM
8x8	5.7884	0.0027

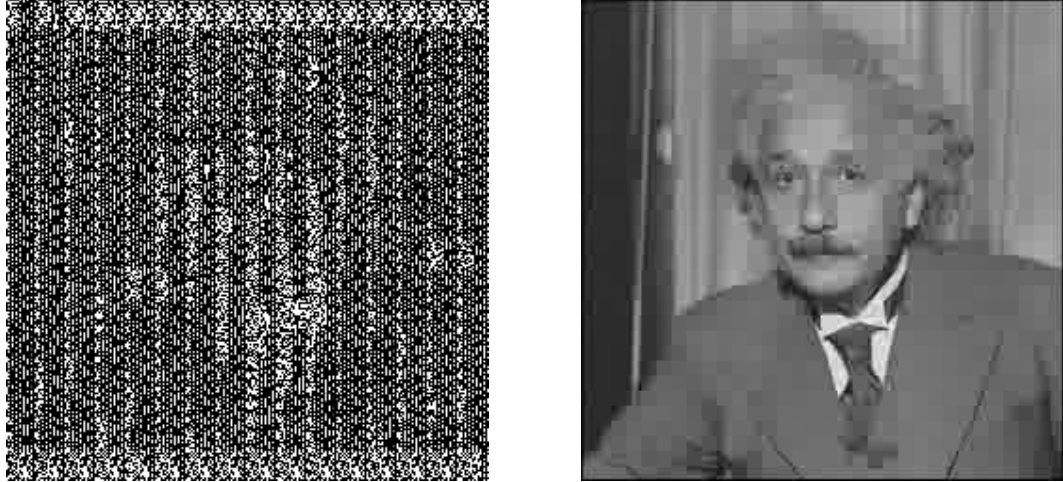


Figure 4.5: Encrypted and Decrypted Einstein Image using DCT with 16x16 Block Size

Figure 4.5 shows both encrypted and decrypted images as results of the proposed DCT algorithm with 16 by 16 block size. Table 4.5 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.5: PSNR and SSIM results of Einstein Image with DCT 16x16 Block Size

Block Size	PSNR	SSIM
16x16	6.1122	0.0057

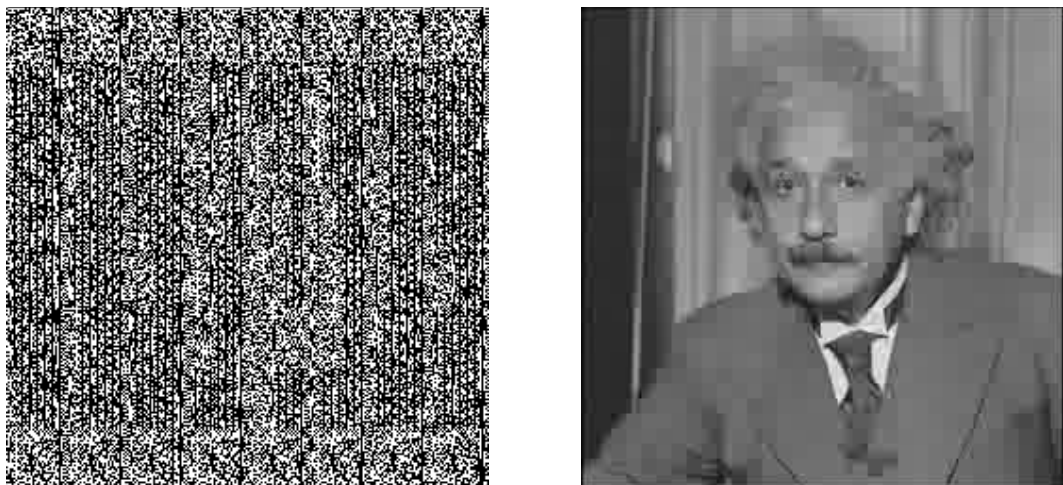


Figure 4.6: Encrypted and Decrypted Einstein Image using DCT with 32x32 Block Size

Figure 4.6 shows both encrypted and decrypted images as results of the proposed DCT algorithm with 32 by 32 block size. Table 4.6 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.6: PSNR and SSIM results of Einstein Image with DCT 32x32 Block Size

Block Size	PSNR	SSIM
32x32	5.7839	0.0059

4.1.3 Peppers Image using DCT Algorithm

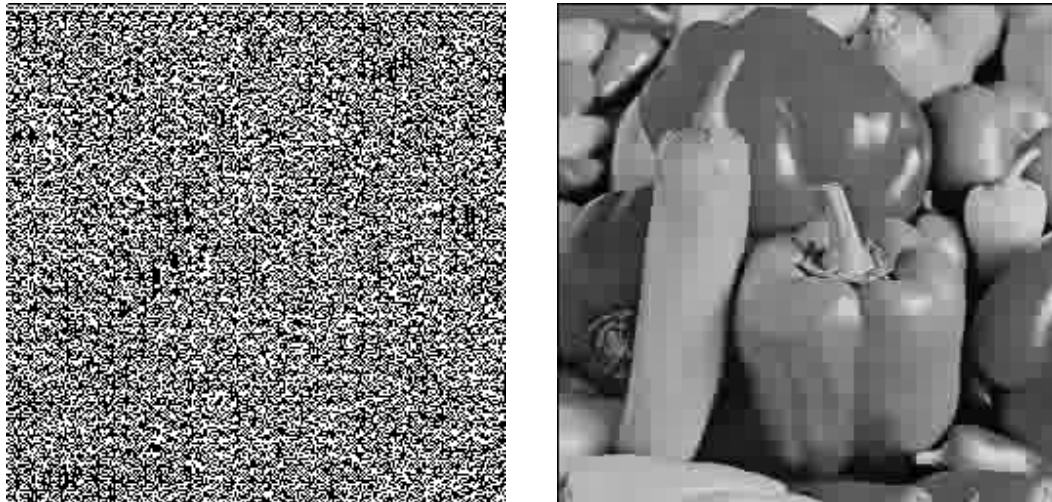


Figure 4.7: Encrypted and Decrypted Peppers Image using DCT and with Block Size

Figure 4.7 shows both encrypted and decrypted images as results of the proposed DCT algorithm with 8 by 8 block size. Table 4.7 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.7: PSNR and SSIM results of Peppers Image with DCT 8x8 Block Size

Block Size	PSNR	SSIM
8x8	5.3789	0.0072

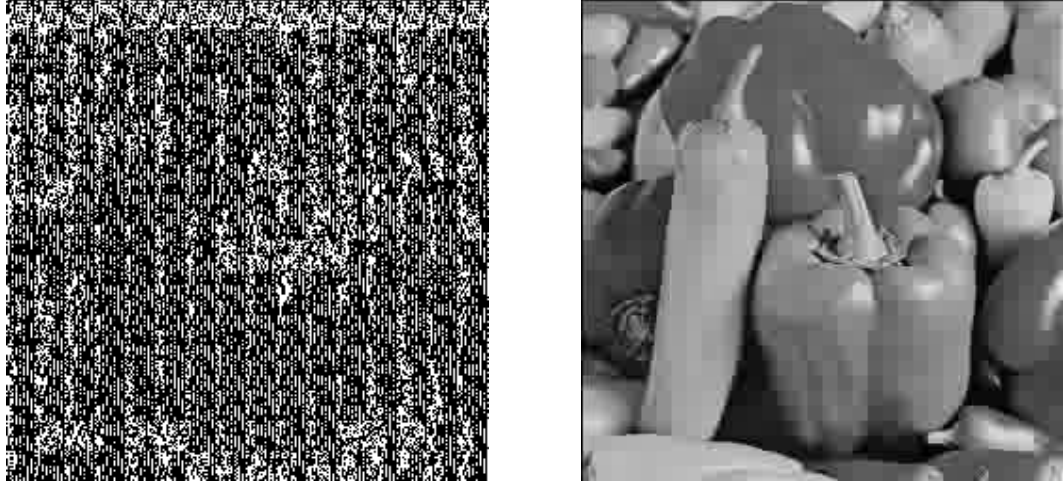


Figure 4.8: Encrypted and Decrypted Peppers Image using DCT with 16x16 Block Size

Figure 4.8 shows both encrypted and decrypted images as results of the proposed DCT algorithm with 16 by 16 block size. Table 4.8 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.8: PSNR and SSIM results of Peppers Image with DCT 16x16 Block Size

Block Size	PSNR	SSIM
16x16	5.6340	0.0090

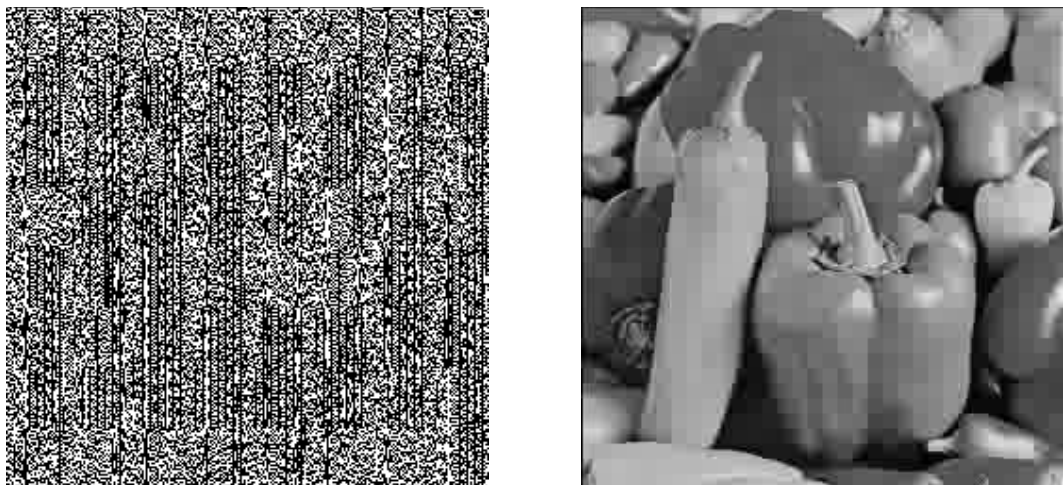


Figure 4.9: Encrypted and Decrypted Peppers Image using DCT with 32x32 Block Size

Figure 4.9 shows both encrypted and decrypted images as results of the proposed DCT algorithm with 32 by 32 block size. Table 4.9 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.9: PSNR and SSIM results of Peppers Image with DCT 32x32 Block Size

Block Size	PSNR	SSIM
32x32	5.4767	0.0076

4.1.4 Cameraman Image using DWT Algorithm

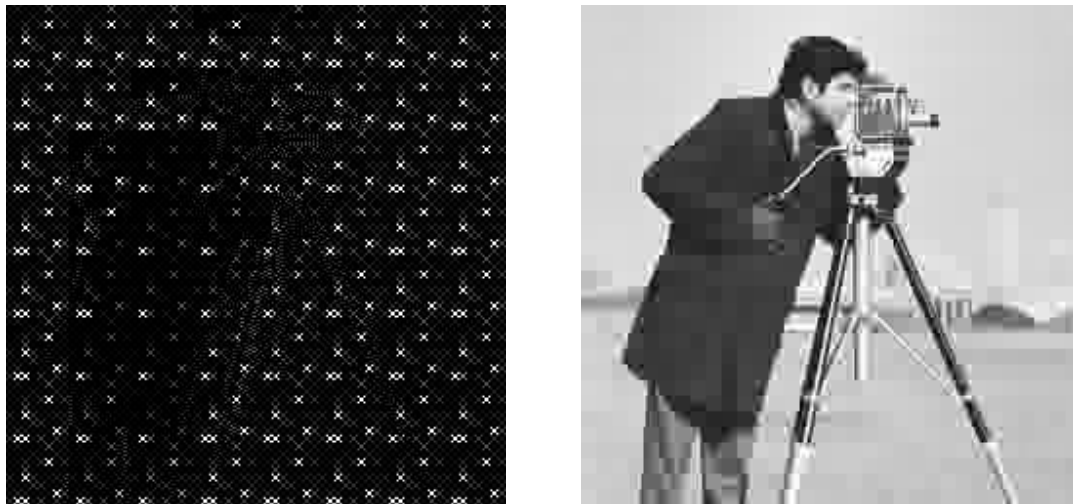


Figure 4.10: Encrypted and Decrypted Cameraman Image using DWT with 8x8 Block Size

Figure 4.10 shows both encrypted and decrypted images as results of the proposed DWT algorithm with 8 by 8 block size. Table 4.10 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.10: PSNR and SSIM results of Cameraman Image with DWT 8x8 Block Size

Block Size	PSNR	SSIM
8x8	3.6236	0.0189

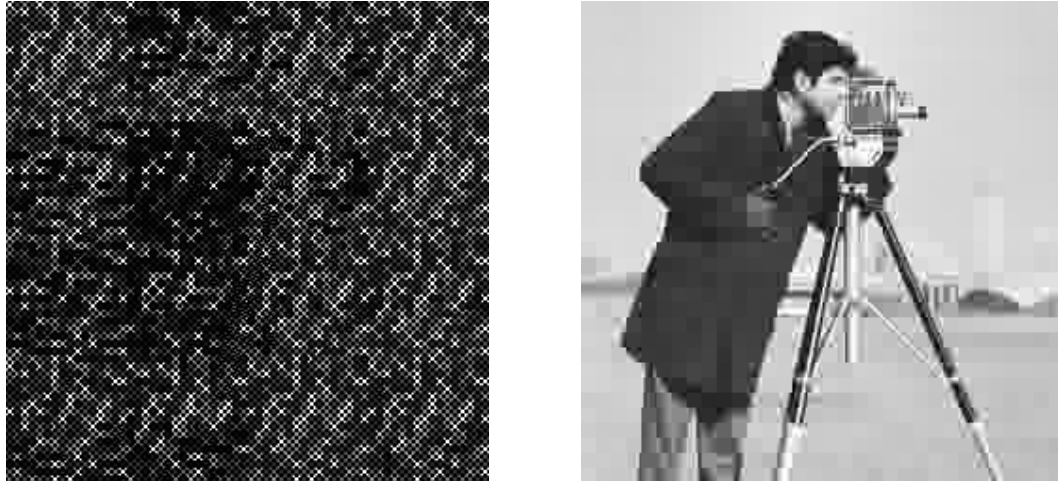


Figure 4.11: Encrypted and Decrypted Cameraman Image using DWT with 16x16 Block Size

Figure 4.11 shows both encrypted and decrypted images as results of the proposed DWT algorithm with 16 by 16 block size. Table 4.11 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.11: PSNR and SSIM results of Cameraman Image with DWT 16x16 Block Size

Block Size	PSNR	SSIM
16x16	4.4578	0.0114



Figure 4.12: Encrypted and Decrypted Cameraman Image using DWT with 32x32 Block Size

Figure 4.12 shows both encrypted and decrypted images as results of the proposed DWT algorithm with 32 by 32 block size. Table 4.12 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.12: PSNR and SSIM results of Cameraman Image with DWT 32x32 Block Size

Block Size	PSNR	SSIM
32x32	4.7881	0.0069

4.1.5 Einstein Image using DWT Algorithm

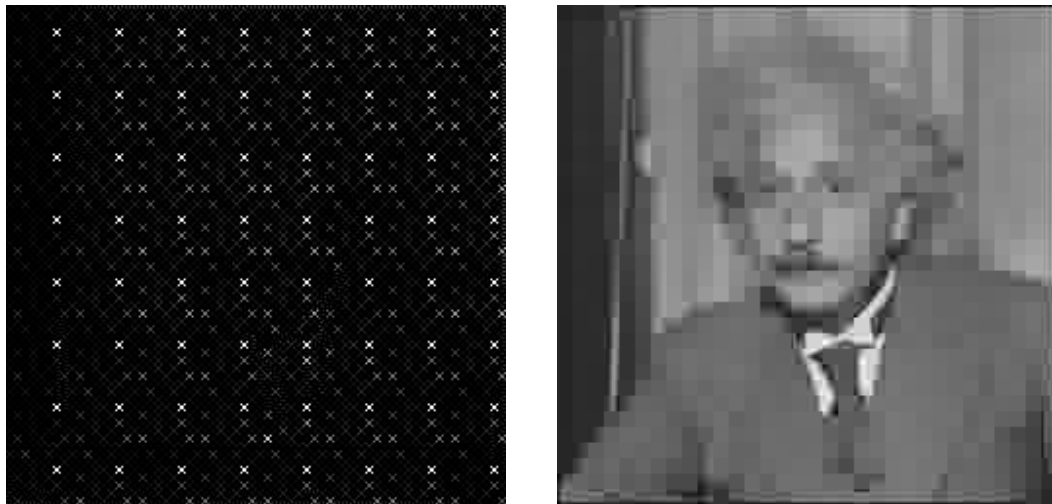


Figure 4.13: Encrypted and Decrypted Einstein Image using DWT with 8x8 Block Size

Figure 4.13 shows both encrypted and decrypted images as results of the proposed DWT algorithm with 8 by 8 block size. Table 4.13 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.13: PSNR and SSIM results of Einstein Image with DWT 8x8 Block Size

Block Size	PSNR	SSIM
8x8	7.5517	0.0273

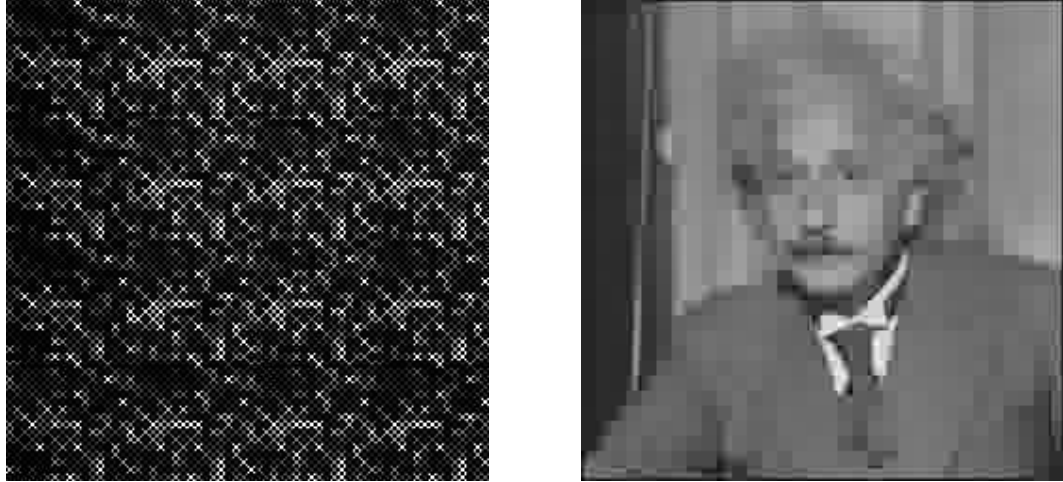


Figure 4.14: Encrypted and Decrypted Einstein Image using DWT with 16x16 Block Size

Figure 4.14 shows both encrypted and decrypted images as results of the proposed DWT algorithm with 16 by 16 block size. Table 4.14 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.14: PSNR and SSIM results of Einstein Image with DWT 16x16 Block Size

Block Size	PSNR	SSIM
16x16	8.2245	0.0181



Figure 4.15: Encrypted and Decrypted Einstein Image using DWT with 32x32 Block Size

Figure 4.15 shows both encrypted and decrypted images as results of the proposed DWT algorithm with 32 by 32 block size. Table 4.15 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.15: PSNR and SSIM results of Einstein Image with DWT 32x32 Block Size

Block Size	PSNR	SSIM
32x32	7.4693	0.0062

4.1.6 Peppers Image using DWT Algorithm

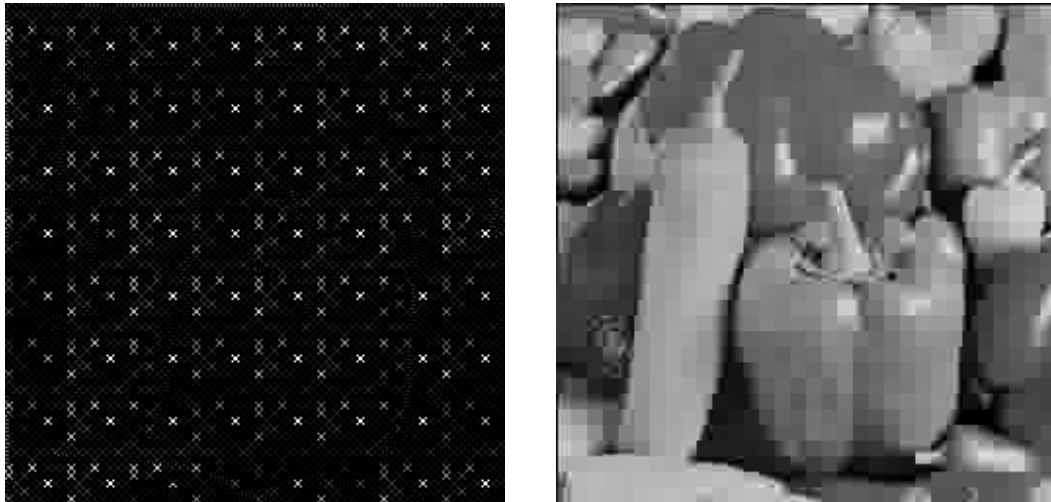


Figure 4.16: Encrypted and Decrypted Peppers Image using DWT with 8x8 Block Size

Figure 4.16 shows both encrypted and decrypted images as results of the proposed DWT algorithm with 8 by 8 block size. Table 4.16 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.16: PSNR and SSIM results of Peppers Image with DWT 8x8 Block Size

Block Size	PSNR	SSIM
8x8	6.6110	0.0256

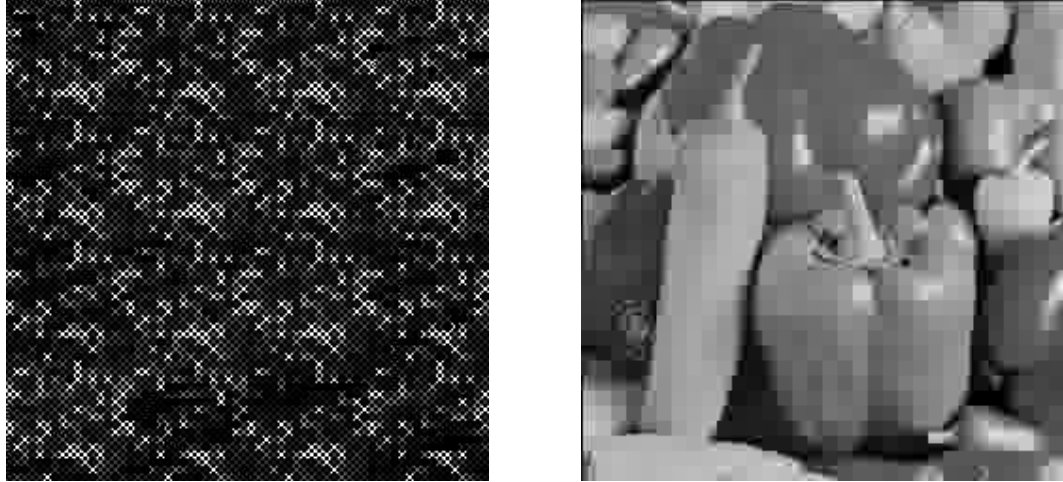


Figure 4.17: Encrypted and Decrypted Peppers Image using DWT with 16x16 Block Size

Figure 4.17 shows both encrypted and decrypted images as results of the proposed DWT algorithm with 16 by 16 block size. Table 4.17 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.17: PSNR and SSIM results of Peppers Image with DWT 16x16 Block Size

Block Size	PSNR	SSIM
16x16	7.1069	0.0159

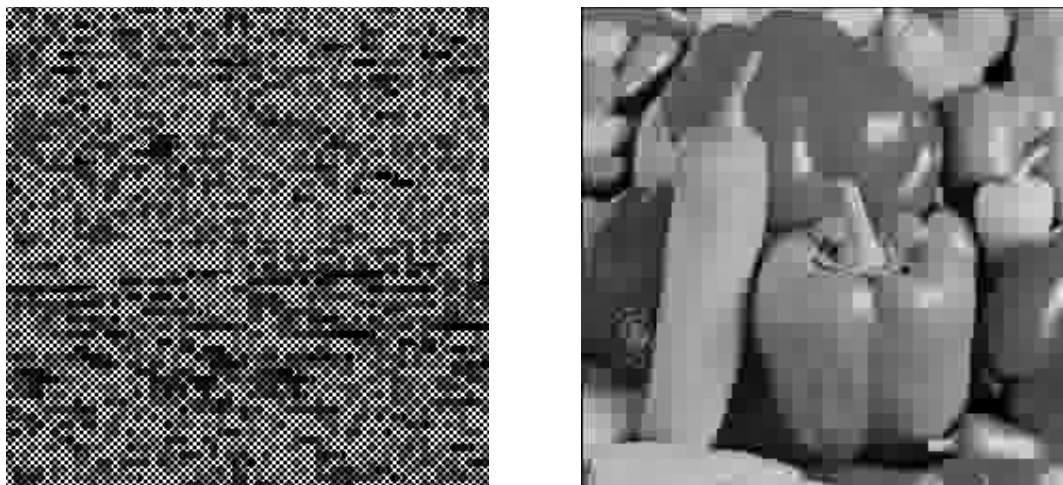


Figure 4.18: Encrypted and Decrypted Peppers Image using DWT with 32x32 Block Size

Figure 4.18 shows both encrypted and decrypted images as results of the proposed DWT algorithm with 32 by 32 block size. Table 4.18 shows the PSNR and SSIM values between the original and encrypted images.

Table 4.18: PSNR and SSIM results of Peppers Image with DWT 32x32 Block Size

Block Size	PSNR	SSIM
32x32	6.7041	0.0048

4.1.7 Comparison Between DCT and DWT Results

Table 4.19 shows the PSNR values for different test results for the proposed DCT and DWT algorithms for each block size 8x8, 16x16 and 32x32.

Table 4.19: PSNR Values of the Proposed Algorithms of the Original and Encrypted Images

	8x8 Block Size		16x16 Block Size		32x32 Block Size	
	DCT	DWT	DCT	DWT	DCT	DWT
Cameraman	4.4987	3.6236	3.8946	4.4578	4.3431	4.7881
Einstein	5.7884	7.5517	6.1122	8.2245	5.7839	7.4693
Peppers	5.3789	6.6110	5.6340	7.1069	5.4767	6.7041

As mentioned in section 3.10.1 the lower PSNR we got between the original and encrypted images the better results we have. So, as we can see in Table 4.19 and Figure 4.19, the PSNR values for DCT are generally better than DWT. In general, both algorithms gave an excellent PSNR results, but we found that the proposed DCT algorithm outperforms the proposed DWT in terms of PSNR values.

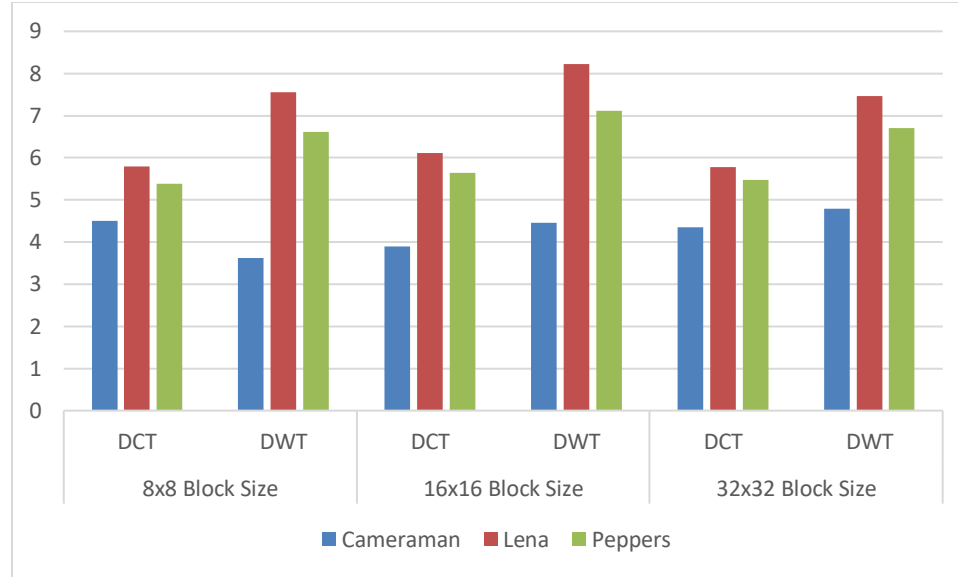


Figure 4.19: PSNR Values of the Proposed Algorithms of the Original and Encrypted Images

To compare between the proposed algorithms in term of SSIM, Table 4.20 shows the SSIM values between the different test cases scenarios, a comparison between the proposed DCT and DWT algorithms for each block size 8x8, 16x16 and 32x32.

Table 4.20: SSIM Values of the Proposed Algorithms of the Original and Encrypted Images

	8x8 Block Size		16x16 Block Size		32x32 Block Size	
	DCT	DWT	DCT	DWT	DCT	DWT
Cameraman	0.0038	0.0189	0.0050	0.0114	0.0041	0.0069
Einstein	0.0027	0.0273	0.0057	0.0181	0.0059	0.0062
Peppers	0.0072	0.0256	0.0090	0.0159	0.0076	0.0048

As mentioned in section 3.10.2 the lower SSIM we got between the original and encrypted images the better results and less similarity we have. So, as we see in both Table 4.20 and Figure 4.20, the SSIM values for DCT are generally much better than DWT. In general, both algorithms gave an excellent SSIM results, but we found that the proposed DCT algorithm outperforms the proposed DWT in terms of PSNR values.

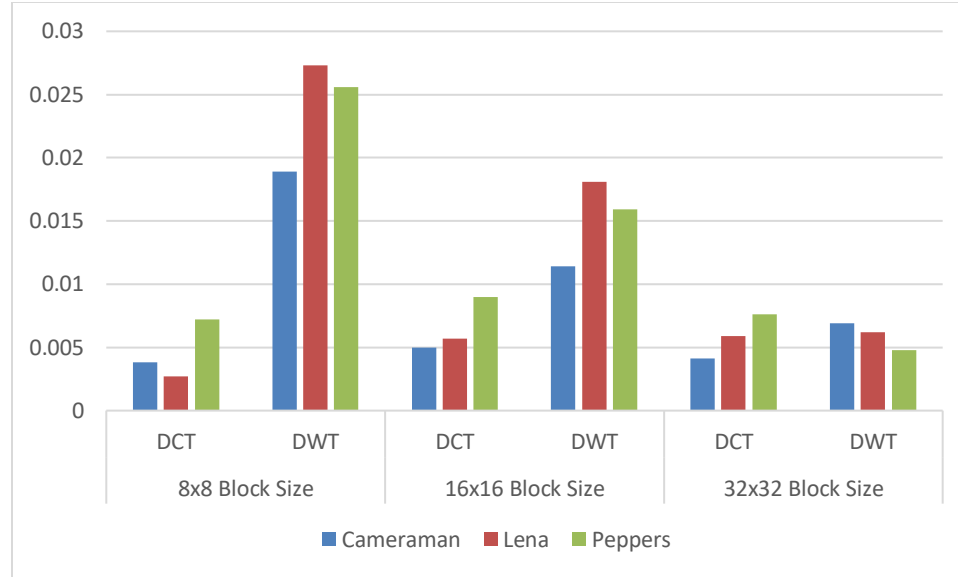


Figure 4.20: SSIM Values of the Proposed Algorithms of the Original and Encrypted Images

As a conclusion, based on the previous results we found that we got the best and lowest similarity and PSNR values between the original and encrypted images are for the proposed DCT algorithm with block size 8x8.

4.2 Histogram Analysis

In this section, we will show the pixel values distribution of the original and encrypted images.

Figure 4.21 shows the plots of images histograms, its clearly shows the pixel values distribution in the original images and how they different from the encrypted images pixel distribution. Also, the encrypted images histograms show approximately equal distribution of pixel values which means that no indication can be observed or any statistical information about the original images.

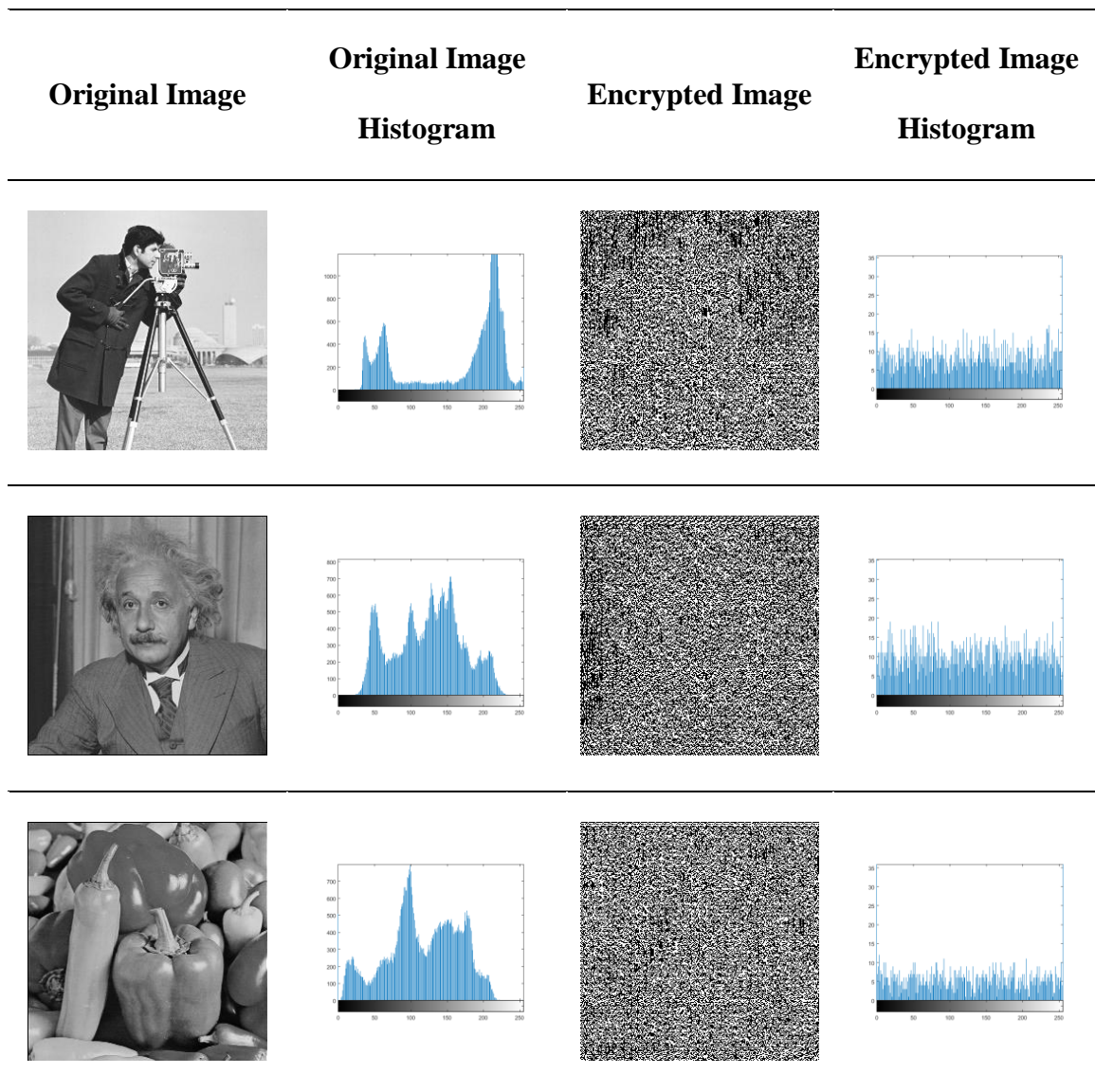


Figure 4.21: Histogram Analysis of DCT Algorithm with 8x8 Block Size

Figure 4.22 shows the plots of images histograms, its clearly shows the pixel values distribution in the original images and how they different from the encrypted images pixel distribution. Also, the encrypted images histograms show approximately equal distribution of pixel values which means that no indication can be observed or any statistical information about the original images.

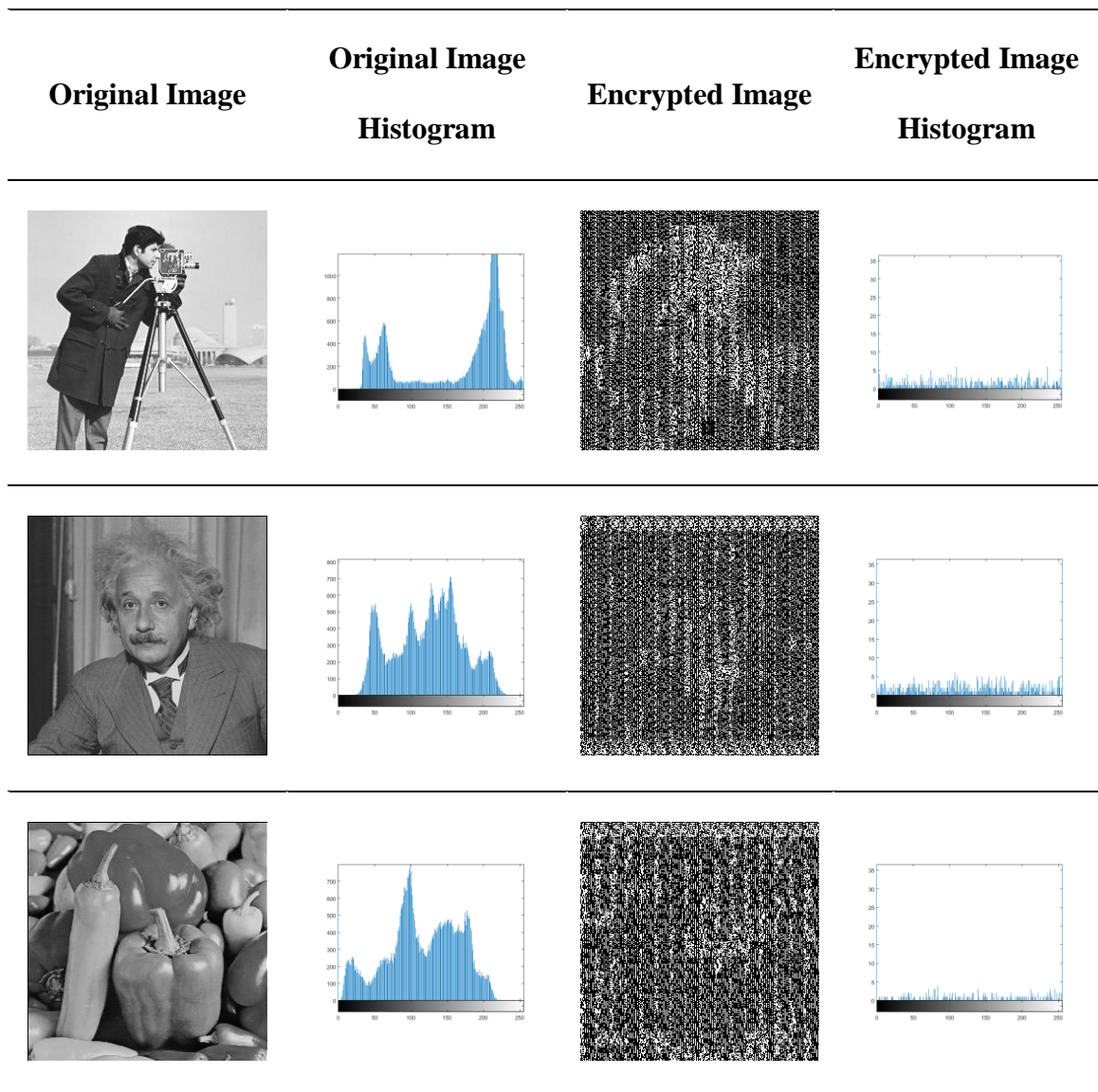


Figure 4.22: Histogram Analysis of DCT Algorithm with 16x16 Block Size

Figure 4.23 shows the plots of images histograms, its clearly shows the pixel values distribution in the original images and how they different from the encrypted images pixel distribution. Also, the encrypted images histograms show approximately equal distribution of pixel values which means that no indication can be observed or any statistical information about the original images.

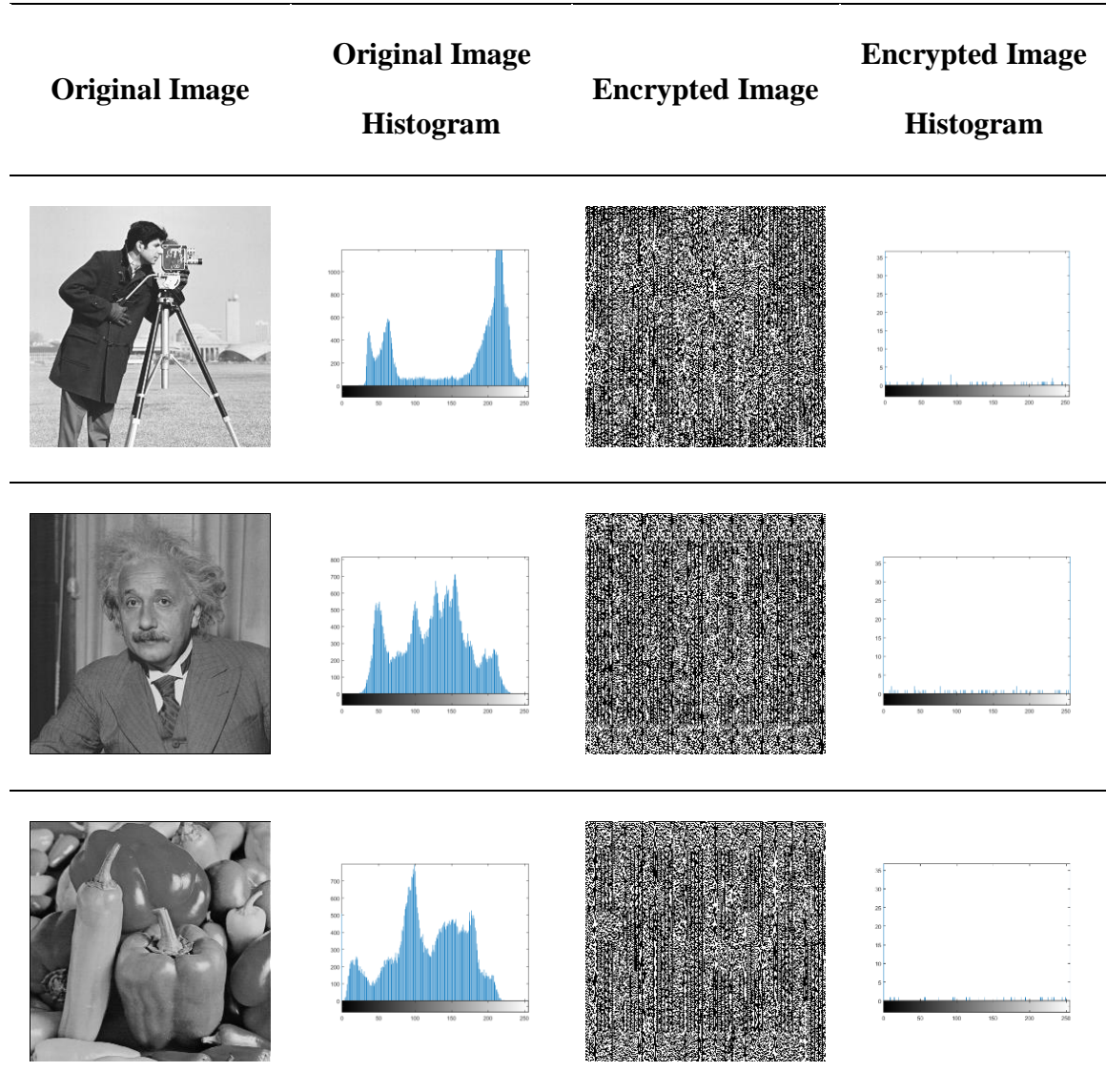


Figure 4.23: Histogram Analysis of DCT Algorithm with 32x32 Block Size

Figure 4.24 shows the plots of images histograms, its clearly shows the pixel values distribution in the original images and how they different from the encrypted images pixel distribution. Also, the encrypted images histograms show approximately equal distribution of pixel values which means that no indication can be observed or any statistical information about the original images.

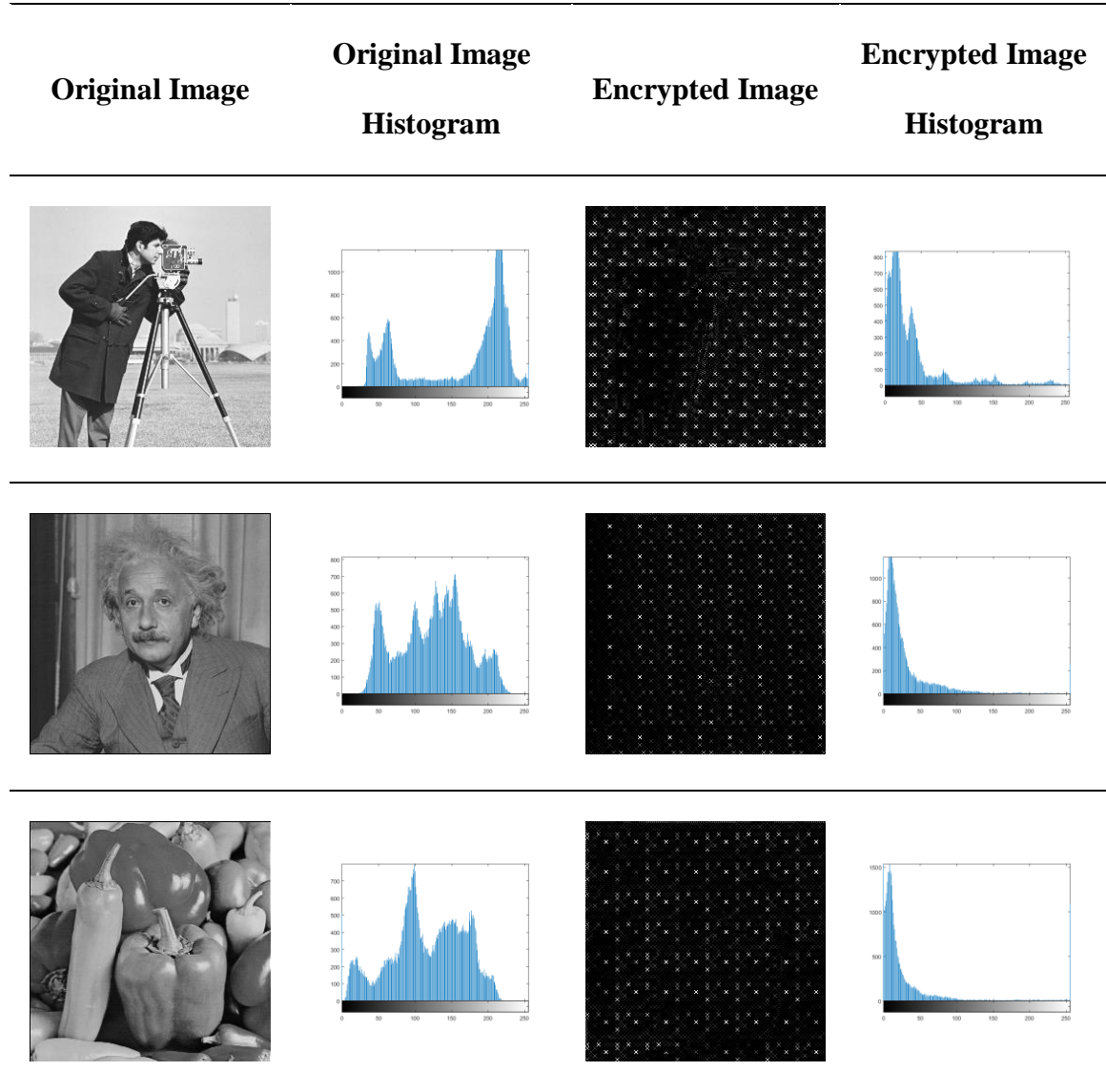


Figure 4.24: Histogram Analysis of DWT Algorithm with 8x8 Block Size

Figure 4.25 shows the plots of images histograms, its clearly shows the pixel values distribution in the original images and how they different from the encrypted images pixel distribution. Also, the encrypted images histograms show approximately equal distribution of pixel values which means that no indication can be observed or any statistical information about the original images.

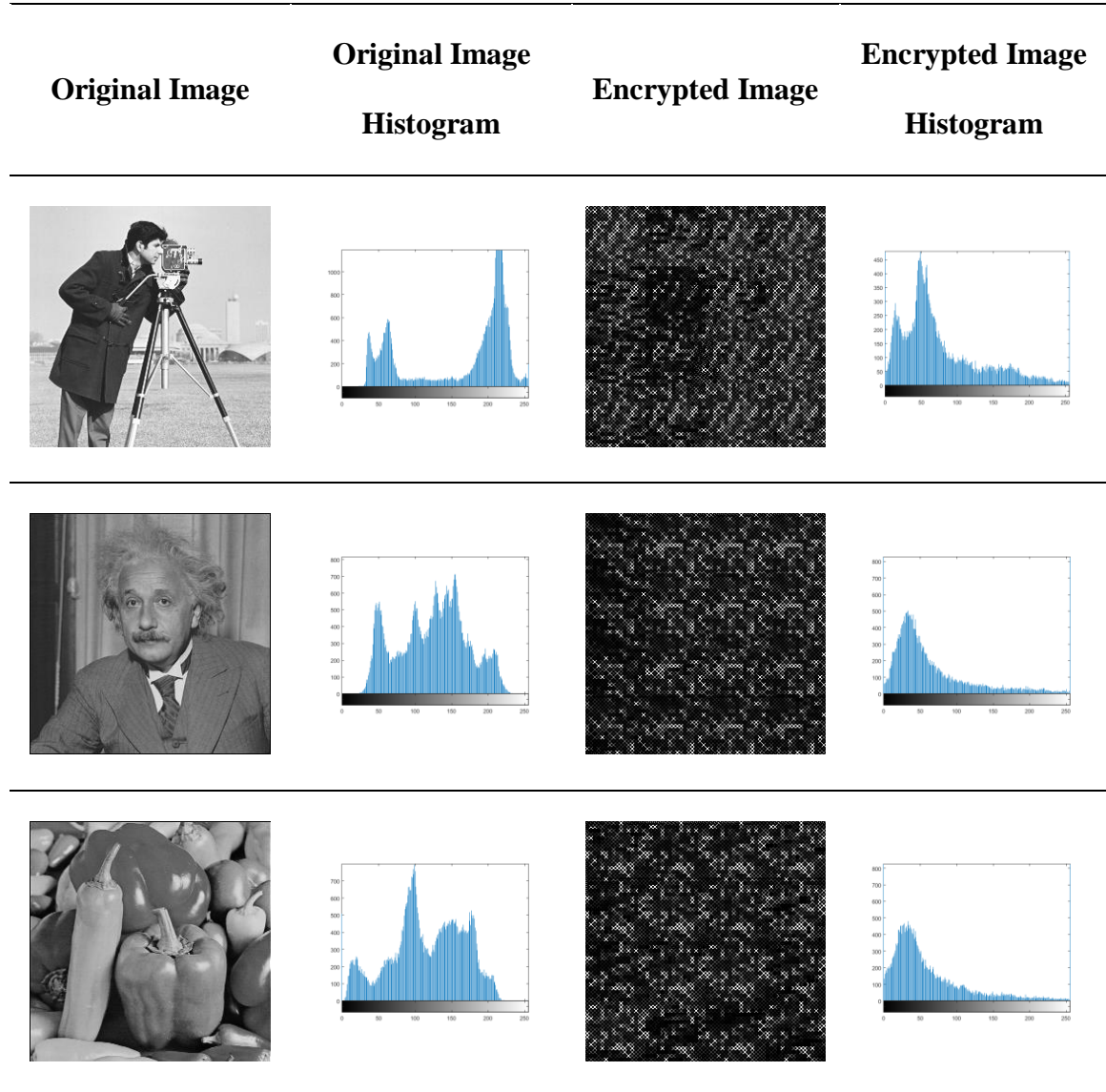


Figure 4.25: Histogram Analysis of DWT Algorithm with 16x16 Block Size

Figure 4.26 shows the plots of images histograms, its clearly shows the pixel values distribution in the original images and how they different from the encrypted images pixel distribution. Also, the encrypted images histograms show approximately equal distribution of pixel values which means that no indication can be observed or any statistical information about the original images.

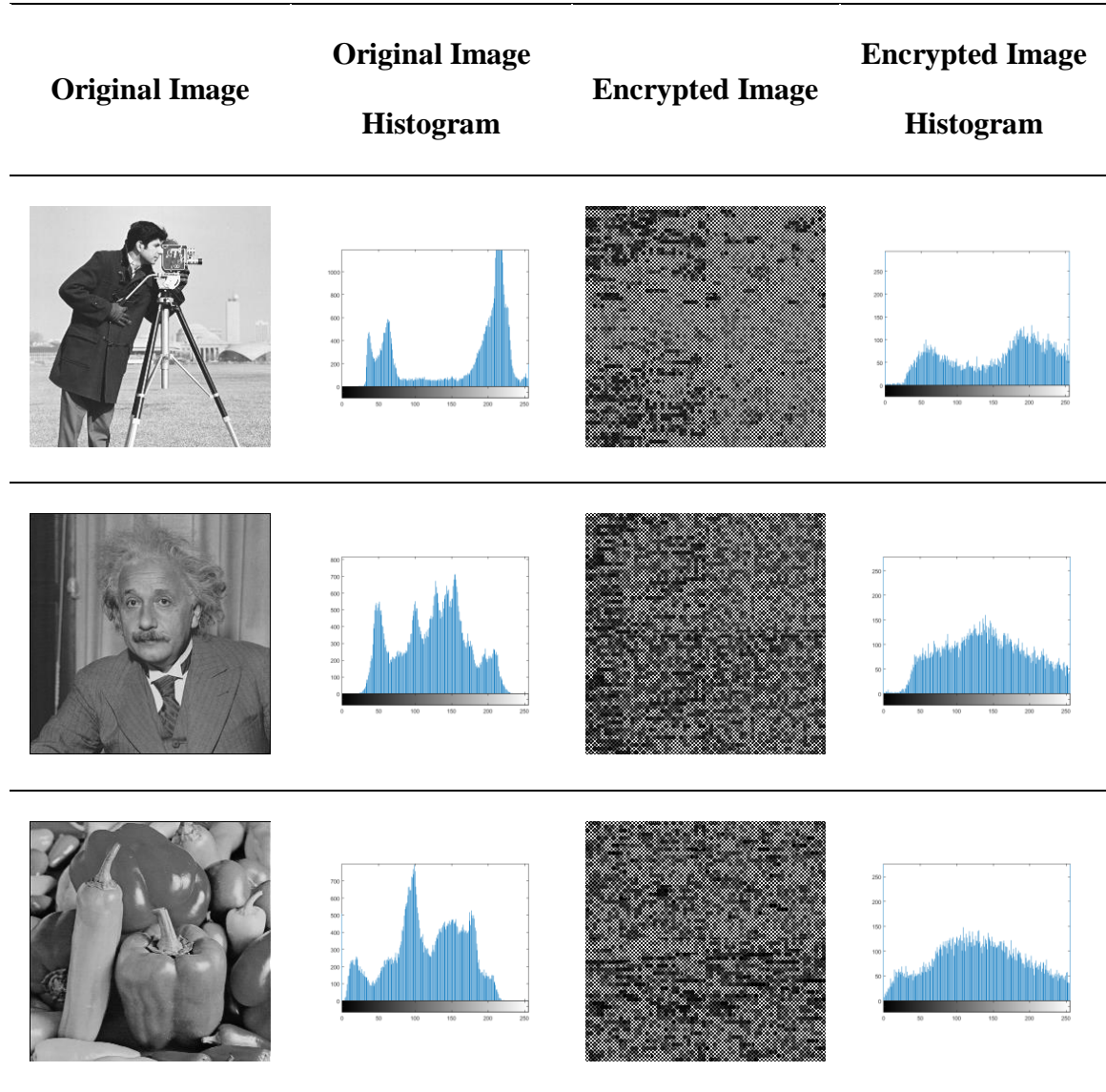


Figure 4.26: Histogram Analysis of DWT Algorithm with 32x32 Block Size

4.3 End-to-End Delay

The end to end delay is the time needed for an image to be encrypted, sent and decrypted back to its original form.

The average time needed for an image to be compressed and encrypted and decrypted was tested, and the results are shown in table 4.21.

Table 4.21: The Average Time for DCT and DWT

	DCT	DWT
Avg. Encryption Time	0.9665 s	0.9141 s
Avg. Decryption Time	0.6086 s	0.5508 s

In Contiki and for the Tmote Sky sensor node each 8768 clock ticks equals to one second, each packet took 0.00015 seconds to be sent through the single hop network and 0.0089 seconds through the multi hop network, the encrypted images files sizes, total number of packets, the required time for each encrypted image to be sent and the total End-to-End delay for both DCT and DWT are shown in tables 4.22, 4.23, 4.24 and 4.25 respectively.

Table 4.22: The Total End-to-End Delay for DCT Through Single-Hop Network

Encryption Technique	Image	Block Size	Original Image Size (Kb)	Compressed Image Size (Kb)	Total Number of Packets	E-to-E Delay (s)
DCT	Cameraman	8x8	64.2	31.6	253	1.6130
		16x16	64.2	31.2	250	1.6126
		32x32	64.2	31.4	252	1.6129
	Einstein	8x8	64.3	38.1	305	1.6208
		16x16	64.3	38.0	304	1.6207
		32x32	64.3	38.4	308	1.6213
	Peppers	8x8	64.6	41.4	332	1.6249
		16x16	64.6	41.4	332	1.6249
		32x32	64.6	41.6	333	1.6251

Table 4.23: The Total End-to-End Delay for DWT Through Single-Hop Network

Encryption Technique	Image	Block Size	Original Image Size (Kb)	Compressed Image Size (Kb)	Total Number of Packets	E-to-E Delay (s)
DWT	Cameraman	8x8	64.2	26.7	214	1.497
		16x16	64.2	26.7	214	1.497
		32x32	64.2	26.7	214	1.4970
	Einstein	8x8	64.3	23.5	188	1.4931
		16x16	64.3	23.4	188	1.4931
		32x32	64.3	23.5	188	1.4931
	Peppers	8x8	64.6	31.8	255	1.5032
		16x16	64.6	31.8	255	1.5032
		32x32	64.6	31.8	255	1.5032

Table 4.24: The Total End-to-End Delay for DCT Through Multi-Hop Network

Encryption Technique	Image	Block Size	Original Image Size (Kb)	Compressed Image Size (Kb)	Total Number of Packets	E-to-E Delay (s)
DCT	Cameraman	8x8	64.2	31.6	253	3.8268
		16x16	64.2	31.2	250	3.8001
		32x32	64.2	31.4	252	3.8179
	Einstein	8x8	64.3	38.1	305	4.2896
		16x16	64.3	38.0	304	4.2807
		32x32	64.3	38.4	308	4.3163
	Peppers	8x8	64.6	41.4	332	4.5299
		16x16	64.6	41.4	332	4.5299
		32x32	64.6	41.6	333	4.5388

Table 4.25: The Total End-to-End Delay for DWT Through Multi-Hop Network

Encryption Technique	Image	Block Size	Original Image Size (Kb)	Compressed Image Size (Kb)	Total Number of Packets	E-to-E Delay (s)
DWT	Cameraman	8x8	64.2	26.7	214	3.3695
		16x16	64.2	26.7	214	3.3695
		32x32	64.2	26.7	214	3.3695
	Einstein	8x8	64.3	23.5	188	3.1381
		16x16	64.3	23.4	188	3.1381
		32x32	64.3	23.5	188	3.1381
	Peppers	8x8	64.6	31.8	255	3.7344
		16x16	64.6	31.8	255	3.7344
		32x32	64.6	31.8	255	3.7344

4.4 Energy Efficiency and Power Consumption

Tmote Sky is powered by two AA batteries. The module was designed to fit the two AA battery form factor. AA cells may be used in the operating range of 2.1 to 3.6V DC and its average current consumption equals to 21 mA . Each AA battery provides a capacity equals to 2400 mA/h.

The battery lifetime is calculated using equation 4.1:

$$Battery\ Lifetime = \frac{Battery\ Capacity\ (\frac{mA}{h})}{Current\ Consumption\ (mA)} \times 0.7 \quad (4.1)$$

Using equation 4.1 we found that the 2xAA batteries life time equals to 160 hours, the following results based on 256x256 pixel images compressed and encrypted using both DCT and DWT algorithms.

- Total number of images that can be sent continuously using DCT algorithm through single-hop network equals to 355,555 images.
- Total number of images that can be sent continuously using DWT algorithm through single-hop network equals to 386,577 images.
- Total number of images that can be sent continuously using DCT algorithm through multi-hop network equals to 127,433 images.
- Total number of images that can be sent continuously using DWT algorithm through multi-hop network equals to 174,545 images.

4.5 Comparison Between the Proposed Algorithms and Others Work

A comparison was conducted using PSNR values between the original and encrypted images, we compared our algorithms with an algorithm proposed by Samson and Sastry [32] and another algorithm proposed by Sethi and Sharma [33] . The results showed that our algorithms outperform others work. The results are shown in table 4.26.

Table 4.26: Comparison Between the Proposed Algorithms and Other Algorithms

	Block Size	The Proposed DCT Algorithm	The Proposed DWT Algorithm	Samson and Sastry Algorithm [32]	Sethi and Sharma Algorithm [33]
Cameraman Image	8x8	4.4987	3.6236	10.2	9.3
	16x16	3.8946	4.4578		
	32x32	4.3431	4.7881		
Peppers Image	8x8	5.3789	6.6110	15.8	7.7
	16x16	5.6340	7.1069		
	32x32	5.4767	6.7041		

Chapter 5

Conclusion and Future Work

5.1 Research Conclusion

In this thesis, we proposed novel digital image encryption techniques. Two algorithms have been designed and implemented for a new digital image encryption techniques. The proposed algorithms use the discrete cosine transform DCT and the discrete wavelet transform DWT. The algorithms are implemented using Matlab and the encrypted images were used to be transmitted through the wireless sensor network. The Contiki OS and its simulator Cooja are used to simulate the WSN environment, the Tmote sky sensor motes are used as the sender, intermediate and receiver.

The performance metrics are: PSNR, SSIM, Histogram Analysis and E-to-E Delay. Three well known standard benchmark images are used for experimental testing: Cameraman, Einstein and Pepper images of size 256x256 pixels. Each image was tested in three phases by dividing it into blocks of sizes 8x8, 16x16 and 32x32.

The experimental results showed that the DWT algorithms outperforms the DCT algorithm. In terms of PSNR between the original and encrypted images the results of DCT was lower than the results of DWT with slight differences between them. In terms of SSIM the results of DCT are also lower than DWT. The End-to-End delay was better and less delay for the DWT algorithm. In both topologies: single and multi-hops the delay results are better for DWT.

Both algorithms gave satisfactory results, but in general the results of DWT algorithm are better than the results of DCT algorithm.

5.2 Future Work

Finally, as is the case for any computational-based research, there is a scope for further enhancement of the study. Therefore, to enhance the research, some recommendations may be addressed as well:

- Extend the proposed algorithms to be used with colored images.
- Implement the proposed algorithms on a real hardware platform such as TelosB motes.
- Enhance the experimental testing on a real large area topology.
- Extend the proposed algorithms and merge between them and any of the encryption techniques in the literature.

References

- [1] A. B. (Nasa A. R. C. Watson, “Image Compression Using the Discrete Cosine Transform,” *Math. J.*, vol. 4, no. 1, pp. 81–88, 1994.
- [2] S. Tedmori and N. Al-Najdawi, “Image cryptographic algorithm based on the Haar wavelet transform,” *Inf. Sci. (Ny)*, vol. 269, pp. 21–34, 2014.
- [3] T. Sivakumar, “A Novel Framework for Image Encryption using Karhunen-Loeve Transform,” vol. 54, no. 2, pp. 1–6, 2012.
- [4] W. Puech, “Image encryption and compression for medical image security,” *2008 1st Int. Work. Image Process. Theory, Tools Appl. IPTA 2008*, 2008.
- [5] S. Tedmori and N. Al-Najdawi, “Lossless image cryptography algorithm based on discrete cosine transform,” *Int. Arab J. Inf. Technol.*, vol. 9, no. 5, pp. 471–478, 2012.
- [6] Y. Zhou, K. Panetta, and S. Agaian, “Image encryption using discrete parametric cosine transform,” *2009 Conf. Rec. Forty-Third Asilomar Conf. Image Process.*, pp. 395–399, 2009.
- [7] M. A. Shoeran and T. Sikha, “Image Encryption and Decryption using Discrete Cosine Transform (DCT),” *Int. J. Electr. Electron. Eng.*, vol. 7, no. 1, pp. 646–654, 2015.
- [8] G. Ren, J. Han, H. Zhu, J. Fu, and M. Shan, “High Security Multiple-image Encryption using Discrete Cosine Transform and Discrete Multiple-Parameter

- Fractional Fourier Transform,” *J. Commun.*, vol. 11, no. 5, pp. 491–497, 2016.
- [9] M. A. B. Younes and A. Jantan, “Image Encryption Using Block-Based Transformation Algorithm,” *Int. J. Comput. Sci.*, vol. 35, no. 1, pp. 407–415, 2008.
- [10] M. Ali, B. Younes, and A. Jantan, “An Image Encryption Approach Using a Combination of Permutation Technique Followed by Encryption,” *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 8, no. 4, pp. 191–197, 2008.
- [11] M. Zeghid, M. Machhout, and L. Khriji, “A modified AES based algorithm for image encryption,” *World Acad. Sci. Eng. Technol.*, vol. 1, no. 1, pp. 70–75, 2007.
- [12] T. Shah, I. Hussain, M. A. Gondal, and H. Mahmood, “Statistical analysis of S-box in image encryption applications based on majority logic criterion,” vol. 6, no. 16, pp. 4110–4127, 2011.
- [13] A. B. Abugharsa, A. H. Basari, and H. Almangush, “A Novel Image Encryption Using an Integration Technique of Blocks Rotation Based on the Magic Cube and the AES Algorithm,” vol. 9, no. 4, pp. 41–47, 2012.
- [14] A. A. Shtewi, B. E. M. Hasan, A. El, and F. A. Hegazy, “An Efficient Modified Advanced Encryption Standard (MAES) Adapted for Image Cryptosystems,” vol. 10, no. 2, pp. 226–232, 2010.
- [15] I. A. Ismail, M. Amin, and H. Diab, “A digital image encryption algorithm based a composition of two chaotic logistic maps,” *Int. J. Netw. Secur.*, vol. 11, no. 1, pp. 1–10, 2010.

- [16] R. Enayatifar and A. Hanan, "Image Security via Genetic Algorithm," vol. 14, pp. 198–203, 2011.
- [17] K. Singh, "Image Encryption using Chaotic Maps and DNA Addition Operation and Noise Effects on it," vol. 23, no. 6, pp. 17–24, 2011.
- [18] Q. H. Alsafasfeh and A. A. Arfoa, "Image Encryption Based on the General Approach for Multiple Chaotic Systems," *J. Signal Inf. Process.*, vol. 2, no. 3, pp. 238–244, 2011.
- [19] M. Prasad and K. L. Sudha, "Chaos Image Encryption using Pixel shuffling," *Comput. Sci. Inf. Technol. (CS IT) CCSEA*, pp. 169–179, 2011.
- [20] X. Huang, "A NEW DIGITAL IMAGE ENCRYPTION ALGORITHM BASED ON 4D CHAOTIC SYSTEM," *Int. J. Pure Appl. Math.*, vol. 80, no. 4, pp. 609–616, 2012.
- [21] H. M. Al-najjar, "Digital Image Encryption Algorithm Based on a Linear Independence Scheme and the Logistic Map," in *The 12th International Arab Conference on Information Technology*, pp. 1–4, 2011
- [22] H. Ogras and M. Turk, "Digital Image Encryption Scheme using Chaotic Sequences with a Nonlinear Function," pp. 555–558, 2012.
- [23] K. Faraoun, "Chaos-based key stream generator based on multiple maps combinations and its application to images encryption," *Int. Arab J. Inf. Technol.*, vol. 7, no. 3, pp. 231–240, 2010.

- [24] S. K. Panigrahy, B. Acharya, and D. Jena, "Image Encryption Using Self-Invertible Key Matrix of Hill Cipher Algorithm," *Image (Rochester, N.Y.)*, no. February, pp. 21–22, 2008.
- [25] B. Acharya, S. K. Panigrahy, S. K. Patra, and G. Panda, "Image Encryption Using Advanced Hill Cipher Algorithm," *ACEEE Int. J. Signal Image Process.*, vol. 1, no. 1, pp. 663–667, 2009.
- [26] P. S. A. Sesha Pallavi Indrakanti, "Permutation based Image Encryption Technique," *Int. J. Comput. Appl.*, vol. 28(8), no. 8, pp. 45–47, 2011.
- [27] I. S. I. Abuhaiba and M. A. S. Hassan, "Image Encryption Using Differential Evolution Approach In Frequency Domain," *Signal Image Process. An Int. Journal(SIPIJ)*, vol. 2, no. 1, pp. 51–69, 2011.
- [28] H. E. H. Ahmed, H. M. Kalash, and O. S. F. Allah, "Encryption Efficiency Analysis and Security Evaluation of RC6 Block Cipher for Digital Images," *2007 Int. Conf. Electr. Eng.*, vol. 3, no. 1, pp. 1–7, 2007.
- [29] T. Sheltami, M. Musaddiqa, and E. Shakshuki, "Data Compression Techniques in Wireless Sensor Networks," *Futur. Gener. Comput. Syst.*, vol. 64, pp. 151–162, 2016.
- [30] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image Vis. Comput.*, vol. 24, no. 9, pp. 926–934, 2006.
- [31] K. Ishiyama, Y. Sugiura, and T. Shimamura, "Optimized Three Scores

Combination for Image Quality Assessment,” in *Asia Pacific Conference on Circuits and Systems*, pp. 5–8, 2016

- [32] C. Samson and V. Sastry, “A Novel Image Encryption Supported by Compression Using Multilevel Wavelet Transform,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 3, no. 9, pp. 178–183, 2012.
- [33] N. Sethi and D. Sharma, “A Novel Method of Image Encryption using Logistic Mapping,” *Int. J. Comput. Sci. Eng.*, vol. 1, no. 2, pp. 115–119, 2012.

VITAE

Name: Ahmad Mohammad Shaheen

Date of Birth: February 25th, 1990

Place of Birth: Amman – Jordan

Nationality: Jordanian

Phones: Mobile 1: 00966 (0) 530374838

Mobile 2: 00962 (0) 799013592

Skype Name: ahmad_shaheen90

Address: Post Address1: P.O. Box 8123, KFUPM, Dhahran 31261, Saudi Arabia

Post Address2: P.O. Box 20541, Amman 11118, Jordan

E-mail: ahmad_shaheen90@yahoo.com

Education:

- **M.Sc. degree:** **Computer Networks – Computer Engineering Dept.**,
King Fahd University of Petroleum and Minerals,
Dhahran, Saudi Arabia (2014 – 2017)
Thesis: Digital Image Encryption Techniques for Wireless
Sensor Networks
- **M.Sc. degree:** **Computer Science**, Al-Balqa' Applied University,
As-Salt, Jordan (2012 – 2015)
Thesis: Lossless Digital Image Encryption using
Karhunen-Loève Transform
- **B.Sc. degree:** **Computer Networks Systems**, Applied Science
University,
Amman, Jordan (2008 - 2012)
**1st rank in this specialization and faculty with an
accumulative average of 91.6% with Honors**
- **High School:** **Scientific Branch**, Modern Systems Schools
Amman, Jordan (2007 - 2008)

Publications:

- Multi-layers Video Steganography: A Novel Technique for Image Hiding,
TRANSACTIONS ON NETWORKS AND COMMUNICATIONS, VOLUME 4, No.
6 (2016)

- Comparison and Analysis Study between AODV and DSR Routing Protocols in VANET with IEEE 802.11b, *Journal of Ubiquitous Systems & Pervasive Networks*, Volume 7, No. 1 (2016) pp. 07-12